A Separate-and-Conquer Algorithm for Learning Multi-Label Head Rules

Ein Separate-and-Conquer Algorithmus zum Lernen von Multi-Label Head Rules
Master-Thesis by Michael Rapp
Date of Submission:

1. Referee: Prof. Dr. Johannes Fürnkranz
2. Referee: Dr. Eneldo Loza Mencía
3. Referee: Dr. Frederik Janssen

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Computer Science
Knowledge Engineering Group

A Separate-and-Conquer Algorithm for Learning Multi-Label Head Rules
Ein Separate-and-Conquer Algorithmus zum Lernen von Multi-Label Head Rules

Submitted Master-Thesis by Michael Rapp

1. Referee: Prof. Dr. Johannes Fürnkranz
2. Referee: Dr. Eneldo Loza Mencía
3. Referee: Dr. Frederik Janssen

Date of Submission:

Thesis Statement pursuant to §22 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form. In the submitted thesis the written copies and the electronic version are identical in content.

Darmstadt, December 22, 2016

_____

(Michael Rapp)

Abstract

In the area of machine learning, multi-label classification is the task of learning a model from training data in order to be able to assign a set of labels to yet unknown instances [Read et al., 2011, Tsoumakas and Katakis, 2006]. This is in contrast to binary or multi-class classification problems, where only single classes are predicted. For example, newspaper articles can often be associated with multiple topics and a single piece of music can belong to more than one genre at once [Godbole and Sarawagi, 2004, Tsoumakas and Katakis, 2006]. As recent studies have revealed, multi-label classification approaches, which are able to take correlations between labels – e.g. subsumptions or exclusions – into account, are expected to achieve better predictive results [Read et al., 2011]. Loza Mencía and Janssen [2015] have recently proposed a separate-and-conquer rule learning algorithm for solving multi-label classification problems, which is able to discover dependencies between individual labels. This is based on inducing rules, which partially or fully depend on label conditions, instead of being exclusively based on testing the values, which are associated with individual instances. The authors of said work advocate the use of rule learning algorithms for solving multi-label classification tasks, because rules are a natural and simple form of expressing a learned model. Furthermore, they allow to expose correlations between labels in a human-readable and -interpretable manner. However, the individual rules, which are learned by Loza Mencía and Janssen's algorithm do only predict the presence or absence of one single label. In order to overcome this restriction, the work at hand aims at modifying the original algorithm in order to be able to induce so-called *multi-label head rules*, which allow to predict multiple labels at once. One challenge of inducing such rules is, that the number of possible label combinations, which have to be taken into account for each rule, exponentially increases with the number of available labels. Therefore, the primary contribution of this work is to elaborate a way for efficiently learning multi-label head rules, even if a large number of labels is given. In order to achieve this, the proposed algorithm is based on exploiting the so-called *anti-monotonicity* of certain evaluation metrics, which are used to measure the performance of potential rules. In this work, it is shown, how said property can be exploited for reducing the computational complexity of searches for multi-label head rules. Furthermore, various evaluation methods, which are commonly used for measuring multi-label classification performance, are examined in order to show, whether they fulfill the properties of anti-monotonicity, or not. This is indispensable for discovering valid configurations of the proposed algorithm and enables to understand its limitations. Finally, by applying the proposed algorithm to different data sets and statistically evaluating the predictions and characteristics of the learned models, the effects of learning multi-label head rules is shown. By comparing the outcome of the proposed algorithm to those of different multi-label classification approaches, it is also shown, that it is able to compete with those approaches in terms of predictive performance.

Zusammenfassung

Auf dem Gebiet des maschinellen Lernens versteht man unter Multi-Label Klassifizierung das Lernen eines Modells auf Basis von Trainingsdaten, um anschließend in der Lage zu sein, eine Menge von Labels noch unbekannten Instanzen zuzuweisen [Read et al., 2011, Tsoumakas and Katakis, 2006]. Dies steht im Gegensatz zu binärer Klassifizierung oder Multiklassen-Problemen, bei denen lediglich einzelne Klassen vorhergesagt werden. Beispielsweise können Zeitungsartikel häufig mit mehreren Themen in Verbindung gebracht werden und ein Musikstück kann zu mehr als einem einzigen Genre gehören [Godbole and Sarawagi, 2004, Tsoumakas and Katakis, 2006]. Wie vergangene Studien zeigten, erzielen Ansätze zur Multi-Label Klassifizierung, die Korrelationen zwischen Labels – z.B. Untergruppen oder gegenseitige Ausschlüsse – berücksichtigen, erwartungsweise bessere Vorhersageergebnisse [Read et al., 2011]. Loza Mencía und Janssen [2015] stellten zuletzt einen Separate-and-Conquer Regellern-Algorithmus zur Lösung von Problemen im Bereich der Multi-Label Klassifizierung vor, der in der Lage ist, Abhängigkeiten zwischen einzelnen Labels aufzudecken. Dies basiert auf dem Lernen von Regeln, die teilweise oder vollständig von Labeln abhängen können, statt ausschließlich die Werte einzelner Instanzen zu berücksichtigen. Die Autoren der genannten Arbeit befürworten den Einsatz von Regellernern zur Lösung von Multi-Label Problemen, da Regeln eine natürliche und simple Form zum Ausdruck eines gelernten Modells darstellen. Außerdem erlauben sie es, Korrelationen zwischen Labeln in einer menschenlesbaren und -interpretierbaren Form aufzuzeigen. Allerdings erlauben die Regeln, die durch den von Menćia und Janssen vorgestellten Algorithmus gelernt werden, lediglich das Vorliegen oder die Abwesenheit eines einzelnen Labels vorherzusagen. Um diese Einschränkung zu überwinden, zielt die vorliegende Arbeit darauf ab, den originalen Algorithmus so zu modifizieren, dass sogenannte *Multi-Label Head Rules*, die die Vorhersage mehrerer Labels erlauben, gelernt werden können. Eine Herausforderung beim Lernen solcher Regeln besteht darin, dass die Anzahl der möglichen Label-Kombinationen, die für jede Regel in Betracht gezogen werden müssen, exponentiell mit der Anzahl der verfügbaren Label ansteigt. Dementsprechend liegt der Beitrag dieser Arbeit in erster Linie darin, eine Möglichkeit zum effizienten Lernen von Multi-Label Head Rules, selbst wenn eine große Anzahl von Labels gegeben ist, zu erarbeiten. Um dies zu erreichen, basiert der vorgestellte Algorithmus auf der Ausnutzung der sogenannten *Anti-Monotonität* bestimmter Evaluationsmetriken, die zur Einschätzung der Güte gelernter Regeln verwendet werden. In dieser Arbeit wird aufgezeigt, wie besagte Eigenschaft ausgenutzt werden kann, um die Komplexität einer Suche nach Multi-Label Head Rules zu reduzieren. Darüber hinaus werden verschiedene Evaluationsmethoden, die üblicherweise für die Einschätzung der Klassifikationsergebnisse bei Multi-Label Problemen verwendet werden, dahingehend untersucht, ob sie die Eigenschaften der Anti-Monotonität erfüllen. Dies ist unabdingbar, um gültige Konfigurationen des vorgestellten Algorithmus aufzuzeigen und erlaubt es, dessen Einschränkungen zu verstehen. Abschließend werden die Auswirkung des Lernens von Multi-Label Head Rules aufgezeigt, indem der vorgestellte Algorithmus auf verschiedene Datensätze angewandt wird und dessen Vorhersagen, sowie die Charakteristika der gelernten Modelle, statistisch untersucht werden. Indem die Ergebnisse des Algorithmus zudem mit denen anderer Ansätze zur Multi-Label Klassifizierung verglichen werden, wird gezeigt, dass er in der Lage ist, mit diesen in Bezug auf die Vorhersagegenauigkeit zu konkurrieren.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

As an introduction to the work at hand, in this first chapter, an overview of the content and structure of the present work should be given. This includes outlining the objectives and challenges of multi-label classification, as well as giving a rough overview of the following chapters' contents.

1.1 Challenges in Multi-Label Classification

The objective of machine learning is to provide automatic, machine-driven tools and algorithms for processing data in order to facilitate categorization, analysis and comprehension [Loza Mencía, 2012]. Classical and well-studied disciplines of this research area are *binary* and *multi-class* classification. Both aim at learning an universally applicable *model* from the assignments between objects and *classes* given in a *training data set*. For example, such objects could represent newspaper articles, of which each one is associated with a single topic such as "politics", "economy", "sports", etc. The model, which is learned by a classifier, should be suited for predicting the class assignments of yet unseen objects. These are given in form of a *test data set*. In order to be able to make predictions on unknown data, the learned model must generalize on the respective type of data. The objects, which are contained in a data set are referred to as *instances* or *examples*. Because in many practical scenarios the individual objects of a data set can be associated with multiple classes at once, *multi-label classification* problems have gained increasing attention in the recent past [Loza Mencía, 2012]. For example, in case of assigning topics to newspaper articles, an individual article can often be categorized by multiple interrelated topics at the same time. For example, the topics "politics" and "economy" could both be associated with an article about reforming tax laws.

Early attempts at solving multi-label classification tasks were based on breaking down the original problem into less complex binary classification problems. These approaches – most notably the *binary relevance* method – are referred to as *problem transformation methods* [Loza Mencía, 2012, Read et al., 2011]. However, studies have revealed that exploiting correlations between labels may be beneficial for the predictive performance (cf. Dembczyński et al. [2012]). Common problem transformation methods do only support this to some extent – either they do not consider correlations between labels at all, or they suffer from high computational complexity when applied to data sets with large numbers of labels. Possible label correlations are implications, subsumptions and exclusions. For example, in the already mentioned scenario of assigning topics to newspaper articles, the topic "foreign affairs" could be a subtopic of "politics". Such subtopics are more likely to be associated with an instance, if the superordinate topic is relevant as well. In order to benefit from the exploitation of label dependencies, while being able to handle a large number of labels at the same time, classification algorithms, which are designed with these goals in mind, are needed.

The exploitation of correlations between labels might not only be desired, because it potentially increases the predictive performance. In some use cases, exposing dependencies between labels could also be required for analyzing multi-label data. In such case, the learned model must be comprehensible and interpretable by humans. In contrast to statistical classification approaches, such as support vector machines or neural networks, rule learning algorithms are well-suited to meet this requirement [Loza Mencía and Janssen, 2015]. As rule learning is one of the oldest and best-researched areas of machine learning, many different strategies and algorithms for the induction of classification rules exist. The rules, which result from the application of such approaches, can not only be used for classifying unknown data, but also form an easily understandable representation of the learned model.

1.2 Organization of the Work

The list, which is given in the following, provides a rough overview of the present work's structure. In order to highlight the main objectives of the next chapters, a brief summary of each of these chapters' content is given.

- **Chapter 2:** In this chapter, the fundamentals of multi-label classification are introduced. This includes a formal definition of the problem domain, as well as a discussion of the most important problem transformation methods (Section 2.1.1) and an argumentation for the need of exploiting label dependencies (Section 2.1.2). Furthermore, the structure of separate-and-conquer rule learning algorithms is introduced (Section 2.2.1) and it is discussed, how they can be used for multi-class classification by using binarization (Section 2.2.2). As the algorithm, which is proposed in this work, is based on the separate-and-conquer algorithm, which was recently proposed by Loza Mencía and Janssen [2015], this also includes a presentation of their work (Section 2.2.3). Finally, the chapter concludes by giving an outlook on different methods, which can be used for measuring the performance of multi-label rules. Besides introducing the mathematical notation, which is used throughout this work (Section 2.3.1), this corresponds to the discussion of different aggregation and averaging strategies (Section 2.3.2), as well as to the definition of commonly used evaluation functions (Section 2.3.3).

- **Chapter 3:** Based on the fundamentals and notations, which are given in Chapter 2, in this chapter, the properties of *anti-monotonicity* (Section 3.2) and *decomposability* (Section 3.3) are formally defined. It is argued, that they can be exploited for pruning searches for multi-label head rules and therefore enable to efficiently deduce such rules with respect to computational complexity. In order to illustrate, how the proposed algorithm searches for multi-label heads, various examples are given in this chapter as well.

- **Chapter 4:** At this point, the operation of the algorithm, which is proposed in the present work, is discussed. As it is based on Loza Menćia and Janssen's separate-and-conquer algorithm, as previously introduced in Section 2.2.3, only aspects that differ from the original approach are considered in this chapter. Most importantly, this includes the refinement of rule conditions (Section 4.1), as well as searching for the best possible multi-label head for an individual rule (Section 4.2). Furthermore, different methods for measuring the performance of multi-label head rules are discussed (Section 4.3) and it is explained, how the rules, which are learned by the algorithm, can be used to classify the label associations of unknown instances (Section 4.4).

- **Chapter 5:** In the individual sections of this chapter, the evaluation functions, which are introduced in Chapter 2, are examined in terms of anti-monotonicity and decomposability. Based on the formal definitions, which are given in Chapter 3, it is mathematically proved or disproved, whether those properties hold for the respective evaluation functions, or not.

- **Chapter 6:** The results of various empirical studies, which have been elaborated as part of this work, are presented in this chapter. By evaluating the performance of the proposed algorithm on different multi-label data sets and comparing the results to those of other multi-label classification approaches, it is possible to gain an impression of the algorithm's capabilities. The comparison of different approaches also includes the binary relevance method and the algorithm for learning single-label head rules by Loza Mencía and Janssen.

- **Chapter 7:** In this final chapter, the content and contributions of the present work are summarized. Furthermore, an outlook on different aspects and enhancements of the proposed algorithm, which may be addressed in the future, is given.

2 Foundations of Multi-Label Classification and Inductive Rule Learning

In this chapter, the fundamentals of multi-label classification are introduced. Besides giving a formal definition of the problem domain, this also includes a discussion of problem transformation methods, which are commonly used for solving such problems, as well as emphasizing the importance of exploiting label correlations. Furthermore – as the algorithm, which is presented in this work, is strongly related to inductive rule learning –, the task of multi-label classification is described with a strong focus on that particular machine learning discipline. Because the algorithm is built on the separate-and-conquer algorithm, which has recently been proposed by Loza Mencía and Janssen [2015], their novel approach for learning multi-label classification rules is also discussed at that point.

2.1 Multi-Label Classification

In machine learning, classification is the task of learning a model from a training data set T. Each of the data set's instances consists of attribute-value pairs and is associated with one or several predefined classes λ_i out of the finite class space $\mathbb{L} := \{\lambda_1, ..., \lambda_n\}$ with $n = |\mathbb{L}|$ being the number of available classes [Loza Mencía, 2012, Loza Mencía and Janssen, 2015]. As each instance X_j assigns concrete values v_k to the corresponding attributes A_k of the given data set, a single instance is defined as follows (cf. Janssen [2012]), where $\mathbb{D}$ denotes the instance space and l corresponds to the total number of attributes. Note, that in this work the terms "instance" and "example" are used as synonyms.

$$X_j := \langle v_1, ..., v_l \rangle \in \mathbb{D} \text{ , with } \mathbb{D} = A_1 \times ... \times A_l \tag{2.1}$$
Instance

Attributes can either be *nominal* or *numerical*. Different types of attributes are possible as well, but they are not relevant for the present work and therefore are not considered here. On the one hand, the values of nominal attributes are discrete – e.g. `true` or `false` in case of a boolean attribute – and cannot be ordered. On the other hand, the values of numerical attributes can be arbitrary numbers out of a continuous value range and therefore are comparable [Fürnkranz, 1999, Janssen, 2012]. As already mentioned, the instances of a data set are associated with classes. Consequently, the training data set T of a classification problem is defined as a sequence of tuples, denoted as follows (cf. Janssen [2012]). In addition to the formal definition, a more descriptive illustration of a data set's structure – according to the notation, which is used throughout this work – is shown in Table 1.

$$T := \langle (X_1, Y_1), ..., (X_m, Y_m) \rangle \subseteq \mathbb{D} \times \mathbb{L} \text{ , with } m = |T| \tag{2.2}$$
Data set

The tuples (X_j, Y_j), which are shown in the equation above, are used to map an individual instance X_j to a corresponding *class vector* Y_j. Such a class vector specifies the classes, which are associated with the instance, by using the following notation (cf. Loza Mencía [2012], Loza Mencía and Janssen [2015]), where each *class attribute* y_i specifies the absence (0) or presence (1) of the corresponding class λ_i:

$$Y_j := \langle y_1, ..., y_n \rangle \in \{0, 1\}^n \text{ , with } n = |\mathbb{L}| \tag{2.3}$$
Class vector

The model, which is derived from a given training data set, can be viewed as the following classifier function, which maps a single instance X to a *prediction* $\hat{Y}$. The prediction is a class vector according to the definition given in Equation 2.3 and therefore the classifier function predicts the classes, which are expected to be associated with an instance. Assuming, that such a classifier function generalizes on the given type of data, it can be used to predict the classes of yet unknown test instances.

$$h(X) = \hat{Y} \tag{2.4}$$
Classifier function

		Attributes						Labels				
		A_1	…	A_k	…	A_l		λ_1	…	λ_i	…	λ_n
Instances	X_1		…		…		Y_1		…		…	
	⋮	⋮		⋮		⋮	⋮	⋮		⋮		⋮
	X_j	v_1	…	v_k	…	v_l	Y_j	y_1	…	y_i	…	y_n
	⋮	⋮		⋮		⋮	⋮	⋮		⋮		⋮
	X_m		…		…		Y_m		…		…	

Table 1: The structure of a multi-label data set according to the notation, which is used throughout this work

In traditional *binary* or *multi-class classification*, each instance is associated with exactly one class and therefore only one class attribute of the class vector, which is predicted by the learned classifier function, is set to 1. If only two predefined classes are available ($|\mathbb{L}| = 2$), the task is considered to be a binary classification problem. Accordingly, if more than two classes are given ($|\mathbb{L}| > 2$), one does refer to such as a multi-class classification task [Loza Mencía, 2012, Tsoumakas and Katakis, 2006]. In contrast, in *multi-label classification*, the instances can be related to an arbitrary number of distinct classes [Godbole and Sarawagi, 2004, Loza Mencía, 2012, Loza Mencía and Janssen, 2015, Tsoumakas and Katakis, 2006]. As a result, there are 2^n potential predictions for an individual instance in such scenario (with $n = |\mathbb{L}|$ being the total number of available classes) [Loza Mencía, 2012]. As this a drastic increase, when compared to the n potential predictions in case of binary or multi-class classification, multi-label classification is a particularly challenging research area. In the context of multi-label classification, "classes" are often referred to as "labels" (cf. Loza Mencía [2012]). Therefore said terminology is used throughout the remainder of the present work. Accordingly, the terms "label vector" and "label attribute" are used instead of "class vector" and "class attribute" in terms of multi-label classification.

2.1.1 Problem Transformation Methods

One possible approach for solving multi-label classification tasks is to use problem transformation methods. Such methods are based on breaking down a complex multi-label classification problem into multiple binary classification problems, which can be solved individually by using common binary classifiers [Loza Mencía, 2012, Read et al., 2011]. The single-class predictions, which are obtained from the binary classifiers, can finally be transformed into multi-label predictions in order to solve the original multi-label classification task [Read et al., 2011]. In the following the most common problem transformation methods are discussed:

- **Binary Relevance:** This is the most common representative of problem transformation methods for solving multi-label classification tasks. It is based on learning a binary classifier for predicting the presence of each label of the original multi-label classification problem. Therefore, a multi-label problem with n labels is decomposed into n binary subproblems [Loza Mencía, 2012, Read et al., 2011]. In order to use the trained binary classifiers to determine the label vector of a yet unknown test instance, the predictions of all of these classifiers have to be queried. As the outcome of each classifier predicts the presence or absence of the corresponding label, the predictions can be transformed into a label vector, which specifies the labels that are assumed to be associated with the given instance.

- **Pairwise Decomposition:** This approach has originally been designed for multi-label ranking (which is not part of this work), but can also be applied as a problem transformation method for solving multi-label problems [Fürnkranz et al., 2008, Read et al., 2011]. As it is based on learning a binary classifier for each pair of labels, the original problem is decomposed into $\frac{n(n-1)}{2}$ subtasks. In order to predict the labels, which are relevant to a test instance, the predictions of all

binary classifiers must be obtained. Each of the obtained predictions can be interpreted as a vote for one of the two corresponding labels [Loza Mencía, 2012]. By aggregating all obtained predictions, the labels can be sorted by their relevance to the given test instance. Finally, by applying a threshold, the labels, which should be included in the final multi-label prediction, can be separated from those, which should not be included.

- **Label Powerset:** When using this problem transformation method, each possible label set is considered as a separate class. Given n labels, 2^n potential label combinations exist. Therefore an original multi-label classification problem is transformed into a multi-class classification task with 2^n classes. The meta classification task can either be solved by using a common multi-class classifier or by further decomposing it into binary classification problems [Loza Mencía, 2012, Read et al., 2011]. In order to classify a test instance, the label set that corresponds to the class, which is predicted by the meta classifier, is used as the resulting multi-label prediction.

Because the number of meta classifiers, which have to be trained, when using the pairwise decomposition or the label powerset method, grows drastically with increasing number of labels, both approaches suffer from bad computational complexity, if applied to data sets with a large number of labels [Read et al., 2011]. In contrast, the binary relevance method is able to handle data sets with a large number of labels. However, it is not able to expose correlations between labels, as it will be discussed in Section 2.1.2 below.

2.1.2 Label Dependencies

When using the binary relevance method (cf. Section 2.1.1) for solving a multi-label classification problem, a binary classifier is trained per label and therefore the labels are implicitly considered to be independent from each other. However, this assumption does not hold for most data sets and it has been shown, that approaches, which are able to exploit dependencies between labels, may benefit from an enhanced classification performance [Dembczyński et al., 2012, Loza Mencía, 2012, Loza Mencía and Janssen, 2015, Read et al., 2011]. For example, imagine a multi-label data set with labels that represent topics of newspaper articles. Given a label λ_u, referring to the topic "politics", and another label λ_v, referring to its subtopic "foreign affairs", it is obvious, that there is a dependency between both labels. I.e., if an instance is associated with λ_u, this implies that label λ_v is present as well (cf. Loza Mencía [2012], Loza Mencía and Janssen [2015]). Unlike the binary relevance method, the label powerset method (cf. Section 2.1.1) is able to model such label dependencies. Nevertheless, it suffers from an exponential computational complexity, depending on the number of labels, and from a tendency towards overfitting. This is due to the fact, that only label sets, which occur in the training data set, are included in the deduced model [Read et al., 2011].

According to Dembczyński et al. [2012], two types of label correlations can be distinguished – namely *conditional* and *unconditional* (or *marginal*) dependencies. Whereas unconditional dependencies do not rely on specific instances, conditional dependencies do depend on the attributes of certain instances [Dembczyński et al., 2012, Loza Mencía, 2012, Loza Mencía and Janssen, 2015]. For example, the dependency between the labels λ_u and λ_v, used in the previously mentioned scenario, is unconditional [Loza Mencía, 2012, Loza Mencía and Janssen, 2015]. When additionally taking into consideration the newspaper article, a specific instance of the data set corresponds to, the probability for the labels to be present has to be assessed differently: On the one hand, if the article's topic is strongly related to politics, the *conditional* probability for both labels – as well as for the dependency between them – to be present increases. On the other hand, if the article is not directly related to politics (e.g. an article about fashion), both labels are very unlikely to be relevant to the given instance and they can be considered to be *conditionally independent*, since the probability for one label to be present is independent of the presence of the other label (cf. Loza Mencía [2012], Loza Mencía and Janssen [2015]).

2.2 Inductive Rule Learning

As the aim of the work at hand is to present a rule learning algorithm, the fundamentals of this machine learning discipline are introduced in this section. The individual rules, which are learned by a rule learning algorithm, consist of a *body*, as well as of a *head*. The following syntax, where the head is denoted as $\hat{Y}$ and the body corresponds to B, is used throughout the remainder of this work (cf. Janssen [2012], Loza Mencía and Janssen [2015]):

$$\hat{Y} \leftarrow B$$

The body of a rule contains one or several conditions, which are used to determine the instances, the rule applies on. These instances are said to be "covered" by the rule (cf. Janssen [2012]). Similar to the publication by Loza Mencía and Janssen [2015], only conjunctive, propositional rules, whose conditions are concatenated using logical `AND` ($\wedge$) operations, are considered in this work. The conditions of a propositional rule are tests on the instances' values, as defined in Equation 2.1. For nominal attributes, equality ($A_k = c_k$) or inequality tests ($A_k \neq c_k$) are used. For numerical attributes, relational tests ($A_k < c_k$, $A_k \leq c_k$, $A_k > c_k$ or $A_k \geq c_k$) are available as well [Fürnkranz, 1999, Janssen, 2012]. Note, that conditions, which perform equality or inequality checks on nominal attributes, are often abbreviated using the notation c_k, respectively $\neg c_k$, in the remainder of this work. The head of a rule specifies the classes, which should be associated with the instances it covers. In case of binary or multi-class classification problems, the head contains a single predictive class attribute ($\hat{y}_i = 0$ or $\hat{y}_i = 1$), which specifies the presence (1) or absence (0) of the corresponding class λ_i [Janssen, 2012]. Similar to the shorthand notation, which is used for denoting tests on nominal attributes, predictive class attributes are often abbreviated using the syntax $\hat{y}_i$ for denoting the presence, respectively $\neg\hat{y}_i$ for denoting the absence, of a class.

	Head	Body	Example
Single-Label Head Rules	positive negative	label-independent	$\hat{y}_1 \leftarrow c_1 \wedge c_2 \wedge c_3$ $\neg\hat{y}_1 \leftarrow \neg c_1 \wedge c_2$
	positive negative	partially label-dependent	$\hat{y}_3 \leftarrow c_1 \wedge \neg\hat{y}_1 \wedge \hat{y}_2$ $\neg\hat{y}_3 \leftarrow \neg c_1 \wedge \neg\hat{y}_1 \wedge \hat{y}_2$
	positive negative	fully label-dependent	$\hat{y}_3 \leftarrow \neg\hat{y}_1 \wedge \hat{y}_2$ $\neg\hat{y}_3 \leftarrow \hat{y}_1 \wedge \neg\hat{y}_2$
Multi-Label Head Rules	sparse dense	label-independent	$\hat{y}_1, \hat{y}_2 \leftarrow c_1 \wedge c_2 \wedge c_3$ $\hat{y}_1, \neg\hat{y}_2, \neg\hat{y}_3 \leftarrow c_1 \wedge c_2 \wedge c_3$
	sparse dense	partially label-dependent	$\hat{y}_3, \hat{y}_4 \leftarrow c_1 \wedge \neg\hat{y}_1 \wedge \hat{y}_2$ $\hat{y}_3, \neg\hat{y}_4, \neg\hat{y}_5 \leftarrow \neg c_1 \wedge \neg\hat{y}_1 \wedge \hat{y}_2$
	sparse dense	fully label-dependent	$\hat{y}_3, \hat{y}_4 \leftarrow \neg\hat{y}_1 \wedge \hat{y}_2$ $\hat{y}_3, \neg\hat{y}_4, \neg\hat{y}_5 \leftarrow \hat{y}_1 \wedge \neg\hat{y}_2$

Table 2: Different types of multi-label rules [Loza Mencía and Janssen, 2015, Table 1]

When inducing rules for handling multi-label classification problems, – depending on the learner's implementation – it is possible to include label conditions in the body of a rule [Loza Mencía and Janssen, 2015, Malerba et al., 1997]. In contrast to *label-independent* rules, such rules allow to expose correlations between labels, as discussed in Section 2.1.2 [Loza Mencía and Janssen, 2015]. If a rule's body exclusively consists of label conditions, it is considered to be *fully label-dependent*. Alternatively, if the body contains label conditions, as well as regular attribute-value tests, the rule is said to be *partially label-dependent*. Furthermore, individual rules may also contain several label assignments in their heads [Loza Mencía and Janssen, 2015]. In contrast to *single-label head rules*, such rules are referred to as *multi-label head rules*. In Table 2 different types of conjunctive rules for multi-label classification are

shown. Whereas the algorithm, which has been proposed by Loza Mencía and Janssen [2015], focuses on inducing single-label head rules, the algorithm, which is presented in this work, aims at learning multi-label head rules as well. Due to the additional expressiveness of such multi-label head rules, it is hoped, that the resulting model is able to better illustrate label correlations. Global dependencies between labels (cf. Section 2.1.2) are best described by using fully label-dependent single-label, respectively multi-label, head rules. Local dependencies can be described by using partially label-dependent bodies instead [Loza Mencía and Janssen, 2015]. Another criterion for categorizing multi-label rules is, whether the presence (denoted as $\hat{y}_i$ in Table 2) or absence (denoted as $\neg\hat{y}_i$ in Table 2) of labels is used in a rule's body or head. Because labels tend to appear relatively infrequently, it is common to only predict the presence of labels and therefore focus on learning rules with *positive* label conditions in their head. In case of multi-label head rules, such rules are also referred to as *sparse*. Nevertheless, in some scenarios, it might be beneficial to learn *negative*, respectively *dense*, rules, for which reason the algorithm, which is proposed in this work – such as its counterpart by Loza Mencía and Janssen [2015] –, is able to induce such rules. Moreover, using conditions, which test the absence of labels, in the body of a fully or partially label-dependent rule, enables to model exclusions in addition to implications and subsumptions.

2.2.1 Separate-and-Conquer Rule Learning

Usually, multiple rules must be learned in order to be able to cover all training examples of the same class (*completeness*), without covering any of those, that are associated with other classes (*consistency*) [Fürnkranz, 1999]. In such case, the resulting rules $r = (r_1, ..., r_n) \in R$ are united as a *rule set* R, which represents the learned model [Loza Mencía and Janssen, 2015]. A widely used strategy for learning rule sets is to use *separate-and-conquer* rule learning algorithms [Fürnkranz, 1999, Janssen, 2012, Janssen and Fürnkranz, 2010]. As pointed out by Fürnkranz [1999], all of these algorithms share the same basic structure, which is shown in Algorithm 1 below.

Require: Training data set $T = \langle (X_1, Y_1), ..., (X_m, Y_m) \rangle$,
class attribute $\hat{y}_i \in \{0, 1\}$, evaluation function δ

1 $R = \emptyset$ ▷ Initialize empty decision list
2 **while** getPositives$(T, \hat{y}_i) \neq \emptyset$ **do** ▷ Learn additional rules until no positive examples remain
3 $(r, T_{covered}) =$ findBestRule$(T, \hat{y}_i, \delta)$
4 $R = R \cup r$ ▷ Add learned rule to decision list
5 $T = T \setminus T_{covered}$ ▷ Remove covered examples from training data set
6 **return** decision list R

Algorithm 1: The basic structure of an iterative separate-and-conquer rule learning algorithm for solving binary classification problems (cf. [Fürnkranz, 1999, Figure 3])

Each separate-and-conquer rule learning algorithm starts with an empty rule set R [Fürnkranz, 1999, Janssen, 2012, Janssen and Fürnkranz, 2010]. As the rules, which are added to the rule set during the execution of the algorithm, are induced successively, it is also referred to as a *decision list* (cf. Janssen [2012], Janssen and Fürnkranz [2010]). According to Algorithm 1, the algorithm takes the training data set T, as well as a class attribute ($\hat{y}_i = 0$ or $\hat{y}_i = 1$) and an evaluation function δ as arguments. The class attribute $\hat{y}_i$ specifies the prediction – i.e. the heads – of the rules, which are induced by the algorithm. Additional rules are learned iteratively by using the subroutine findBestRule. Whenever a new rule is learned, it is added to the decision list and the training examples, which are covered by the rule, are removed from the original training data set. The induction of new rules is continued until all examples in T, for which the class attribute $\hat{y}_i$ is true, are covered [Fürnkranz, 1999]. In order to classify a test example, the rules of the learned decision list are processed in order of their induction. The head of the first rule, which covers the respective example, is used for predicting its class.

Require: Training data set T, class attribute $\hat{y}_i$, evaluation function δ

1 $r = \hat{y}_i \leftarrow \emptyset$ ▷ Start with most general rule
2 $r_{best} = r$
3 $T_{covered} = T$
4 **while** getNegatives$(T_{covered}, \hat{y}_i) \neq \emptyset$ **do** ▷ Refine rule as long as any negatives are covered
5 **for each** possible condition c **do**
6 $r_{refined}.body \cup c$ ▷ Add condition to rule's body
7 **if** evaluateRule$(r_{refined}, T, \delta) >$ evaluateRule(r_{best}, T, δ) **then**
8 $r_{best} = r_{refined}$
9 $r = r_{best}$
10 $T_{covered} =$ getCovered(r, T)
11 **return** best rule r, covered training examples $T_{covered}$

Algorithm 2: Algorithm findBestRule for inducing a rule, based on the current data set (cf. [Fürnkranz, 1999, Figure 3])

According to Algorithm 2, whenever a new rule is about to be induced, it is initialized with an empty body and therefore initially covers all of the training examples. As long as a rule still covers examples, for which the class prediction $\hat{y}_i$ is wrong, the rule is specialized by adding additional conditions to its body [Fürnkranz, 1999, Janssen, 2012, Janssen and Fürnkranz, 2010]. This results in fewer examples being covered by the rule. The possible conditions are attribute-value tests, made up from all available attributes and their corresponding values, as present in the training data set [Fürnkranz, 1999]. Among all refinements of the original rule, the one, which optimizes the given evaluation function δ – e.g. the percentage of correctly classified examples among all covered examples –, is considered to be the best choice [Fürnkranz, 1999, Janssen, 2012, Janssen and Fürnkranz, 2010]. In Section 2.3.3 several evaluation functions, which are relevant to the present work, are introduced. Finally, the subroutine findBestRule returns the induced rule in order to add it to the decision list and causing all training examples it covers to be removed from the current training data set, before the separate-and-conquer algorithm continues with the next iteration.

In order to prevent *overfitting* – a situation where the learned model just reflects the given training examples and does not generalize on unseen data [Janssen, 2012] –, the completeness and consistency constraints are usually relaxed. By either stopping the refinement of rules according to some stopping criterion (*pre-pruning*), or by post-processing the induced rules (*post-pruning*), the inclusion of too specific rules into the decision list can be avoided. This results in learning a more compact model, which is neither complete, nor consistent, but is expected to be more predictive on yet unknown test examples [Fürnkranz, 1999]. Moreover, it is also possible to use a bottom-up strategy, instead of the top-down search, which is given in Algorithm 1. When using such a strategy, each new rule is initialized to cover exactly one of the given training examples. In order to cover additional examples, it is iteratively generalized by removing conditions from its body [Fürnkranz, 1999, Janssen, 2012].

2.2.2 Class Binarization

By default, the separate-and-conquer algorithm, which is discussed in Section 2.2.1, can only be used for handling binary classification problems. This is, because it is only able to discriminate between two classes. Given such a binary classification task, the less frequent of both available classes is usually used as the target class for rule induction using the algorithm. When classifying an unknown test example using the learned model, all rules of the decision list are applied to the example successively. As soon as the first rule covers the example, its head is used for predicting the associated class. The remaining rules must not be processed any further. If none of the decision list's rules covers the test example, a *default rule* applies, predicting the more frequent class, no rules have been learned for [Janssen and Fürnkranz,

2010]. In order to be able to handle multi-class classification problems by using the discussed algorithm, *class binarization* is usually used [Fürnkranz, 2002, Janssen and Fürnkranz, 2010]. The most common binarization techniques are discussed in the following:

- **(Unordered) 1-vs-all Class Binarization:** Given n predefined classes $\lambda_1, ..., \lambda_n$, when using this binarization technique, a multi-class classification problem is decomposed into n binary subproblems. As the classes are not ordered in any way, this method is considered to be "unordered". For each class λ_i, the separate-and-conquer algorithm given in Algorithm 1 is executed using the class argument $\hat{y}_i = 1$. This causes the rules, which are induced by each meta classifier, to cover the examples, which are associated with the current class λ_i, while considering all other examples as negatives. Finally, all learned rules are included in a joint rule set, which is used as a model for the original multi-class classification problem. Because the subproblems are solved in an undetermined order, the rules in the resulting rule set are unordered and contradictive rules – i.e. rules that predict different classes for the same test example – may exist. As a result, when classifying an unseen example, all covering rules must be taken into consideration and conflicts have to be resolved by using some kind of voting mechanism.

- **Ordered 1-vs-all Class Binarization:** This binarization technique is very similar to unordered 1-vs-all class binarization, as described above, except that the classes are ordered by ascending frequency. At first the separate-and-conquer algorithm is applied to the whole training data set, considering the examples of the least frequent class λ_1 to be positives and those of the remaining classes $\lambda_2, ...\lambda_n$ to be negatives. Afterwards, all training examples, which are associated with the already considered class λ_1, are removed from the data set and the procedure is continued by using the second-least frequent class λ_2 as the target class for applying the separate-and-conquer algorithm again. Finally, the rules, which have been learned for each subproblem, are united in an ordered rule set. For predicting the most frequent class, a default rule can be used. Hence no rules, which cover that particular class, must be learned explicitly.

- **Pairwise Class Binarization:** As its name indicates, this binarization method is based on training a classifier for each pair of classes, while leaving out the training examples, which are not associated with either of both classes [Fürnkranz, 2002, Janssen, 2012]. As a result, an original multi-class classification problem with n classes is decomposed into $\frac{n(n-1)}{2}$ subproblems [Janssen, 2012]. Although way more iterations are necessary to solve a classification task that way, it has been shown, that this approach can outperform 1-v-all binarization variants, when implemented efficiently [Fürnkranz, 2002]. The predictions of the trained classifiers can be considered as a vote for one of the two corresponding classes. Therefore, when classifying an unknown test example, the predictions of all classifiers have to be obtained and must be transformed into a multi-class prediction by utilizing a voting mechanism [Fürnkranz, 2002].

By using one of the previously discussed binarization techniques, the basic separate-and-conquer algorithm, which is given in Algorithm 1, can flexibly be adapted for handling multi-class classification problems. However, a different approach for performing binarization is needed in case of multi-label classification. Due to its relevance to the present work, the particular binarization strategy, which has been elaborated by Loza Mencía and Janssen [2015] as part of their work, is discussed in the following section.

2.2.3 An Algorithm for Multi-Label Rule Learning

In this section the algorithm, the present work is based on, is discussed in detail. Both, the original algorithm, as well as the modified version presented in this work, share some similarities and understanding the operation of the original approach is crucial for the remainder of this work. If not marked differently, all information given in this section is taken from the publication by Loza Mencía and Janssen

[2015], the original algorithm has been proposed in for the first time. Said algorithm allows to induce single-label head rules as shown in Table 2 at the beginning of Section 2.2. As it was designed with the fundamentals of separate-and-conquer rule learning algorithms in mind, it is based on the structure previously discussed in Section 2.2.1. However, there are some differences when compared to separate-and-conquer algorithms for solving binary or multi-class classification tasks:

- **Multi-Label Decision Lists:** The algorithm learns a model in form of a decision list – i.e. for classification, the induced rules must be processed in a determinate order. When using a decision list for single-class classification, as previously discussed in Section 2.2.1, the prediction of the first covering rule is used and all other rules must not be taken into consideration. However, in multi-label classification using single-label head rules, the predicted label vector of a test example must be composed of multiple rules' predictions, because each rule does only predict the presence or absence of a single label. As a result, in such case, the classification process is not stopped when an example is covered by a rule unless all of the example's labels are already predicted. Labels, which remain unset after processing all rules, are considered to be irrelevant. This corresponds to the concept of default rules, which is used when learning traditional decision lists for binary and multi-class classification. If a particular label is predicted differently by multiple rules, the prediction of the first covering rule is assumed to be the correct one and its prediction cannot be revoked by other rules afterwards. Additionally, it is possible to mark individual rules as *stopping rules*. Whenever such a rule is encountered, the classification of the current test example is stopped. This may be useful to prevent the prediction of too many relevant labels and is discussed in a more detailed manner in the course of this section.

- **Re-inclusion of Training Examples:** All separate-and-conquer rule learning algorithms are based on iteratively removing covered examples from the training data set once a new rule is induced. However, the algorithm, which is discussed in this section, uses a more complex strategy for removing training examples. This is, because usually multiple single-label head rules have to be learned in order to be able to model an example's full label vector. Consequently, even if a training example is already covered by an induced rule, it must be retained for learning additional rules. Only when the labels, which are associated with an example, are covered to at least some extent, the respective example can be removed from the training data set.

Algorithm 3 illustrates how a multi-label decision list is learned in order to model the label assignments of a training data set T. According to Equation 2.2, each entry of the data set is a tuple, consisting of an instance X_j and a corresponding label vector Y_j. Instead of exposing the true label vectors to the training algorithm, a copy $T_{current}$ of the given data set, with all labels set to be unknown, is created initially (cf. Algorithm 3, line 2). After a new rule has been induced by the algorithm, the label, which is predicted by the new rule, is added to the initially empty data set. The labels, which are marked as already predicted, can be used in later iterations of the algorithm for learning label-dependent rules as shown in Table 2. Besides the training data set, Algorithm 3, also takes targets G as a parameter. Targets allow to specify, whether the induced rules should only predict the presence of labels (if $G = \{1\}$), or if rules are also allowed to predict the absence of labels (if $G = \{0, 1\}$). In each of the algorithm's iterations the subroutine FINDBESTGLOBALRULE is used to learn a new single-label head rule, covering some of the training examples left in the current data set. The newly learned rule is then added to the decision list and the examples it covers are either removed or re-included into the learning process, depending on the results of the subroutines GETCOVEREDSETS and GETREADDSET. The re-inclusion of training examples depends on the parameter τ, as well as on whether stopping rules should be used and whether examples, whose label vectors are fully predicted by already induced rules, should be re-added. The induction of additional rules is continued until only θ examples are left in the data set. Finally, the decision list R is returned. It contains all rules, which have been induced during the algorithm's execution, and represent the learned model.

Require: Training data set $T = \langle (X_1, Y_1), ..., (X_m, Y_m) \rangle$,
parameters θ, τ, evaluation function δ, targets G (either $G = \{1\}$ or $G = \{0, 1\}$),
whether using stopping rules, whether re-inserting fully covered examples

$R = \emptyset$ ▷ Initialize empty multi-label decision list
$T_{current} = \langle (X_1, Y_1), ..., (X_m, Y_m) \rangle$ with $X_j \in T$ and $Y_j = (?, ..., ?)$, $j = 1...m$
while $|T|/m \geq \theta$ **do** ▷ Until, e.g., 95% of examples covered
  $r =$ findBestGlobalRule$(T, T_{current}, G, \delta)$ ▷ Get best possible rule regardless the head
  $R = R \cup r$ ▷ Add rule to decision list
  $(T, T_{part}, T_{full}) =$getCoveredSets$(T, T_{current}, r)$ ▷ Separate T according covering by r
  $T_{add} =$ getReAddSet(T_{part}, T_{full}) ▷ Depending on user parameters
  if $T_{add} = \emptyset$ **then**
    mark r as stopping rule ▷ Only uncovered examples in T of next round
  else
    $T = T \cup T_{add}$ ▷ Add also some covered examples, do not remove them
return multi-label decision list R

Algorithm 3: Separate-and-conquer algorithm for learning single-label head rules [Loza Mencía and Janssen, 2015, Fig. 3]

The subroutine findBestGlobalRule, which is shown in Algorithm 4, is used to induce a new single-label head rule in each one of the algorithm's iterations. It first learns a rule for each label and target and finally chooses the best rule among all labels, according to a performance measurement using the evaluation function δ and the true label vectors of the original training data set. In order to search for the best rule given a particular label-target combination, the subroutine findBestRule may be implemented as a top-down or bottom-up search similar to the example shown in Algorithm 2. Due to the re-inclusion of training examples, the same training examples may be used in subsequent iterations of the algorithm. Without proper handling of such cases, the same rule as before would be returned by the subroutine findBestRule, resulting in the training data set to not be altered for the next iteration either. In order to prevent such infinite loops, training examples, for which the current label is marked to be already predicted in $T_{current}$, are not exposed to the subroutine findBestRule.

Require: Original training data set T, current training data set $T_{current}$,
targets G, evaluation function δ

$r_{best} = \emptyset \leftarrow \emptyset$
for each possible label attribute $\hat{y}_i \in G$ **do** ▷ Find best rule for each label and target
  $T_i = T \setminus$ all X_j where $Y_i \in T_{current}$ is already set ▷ Remove examples with label already predicted
  $(r, T_{covered}) =$ findBestRule$(T_i, \hat{y}_i, \delta)$ ▷ Find best body for target $\hat{y}_i \in G$
  if evaluateRule$(r, T_i, \delta) >$ evaluateRule(r_{best}, T_i, δ) **then**
    $r_{best} = r$ ▷ Replace by better rule
return best rule r_{best}

Algorithm 4: Algorithm findBestGlobalRule for inducing a single-label head rule, based on the current data set [Loza Mencía and Janssen, 2015, Fig. 4]

Whenever a new rule is learned, the subroutines getCoveredSets and getReAddSet are used in order to decide, whether the examples, which are covered by the rule, should be removed from the training data set, or re-added instead. The subroutine getCoveredSets, as shown in Algorithm 5, updates the labels, which are predicted by a newly learned rule, in the data set $T_{current}$ and removes all covered examples from the original training data set T. Depending on whether their label vectors are partially or fully set, the instances are either added to the data set T_{part} or T_{full}.

Require: Original training data set T, current training data set $T_{current}$, rule r

$T_{part} = \emptyset, T_{full} = \emptyset$
for each example $(X_j, Y_j) \in$ getCovered$(r, T_{current})$ **do** ▷ Compute covering status for each example
  apply $r.head$ on Y_j ▷ Add prediction of r to corresponding label vector in $T_{current}$
  if Y_j is fully set **then**
    $T_{full} = T_{full} \cup (X_j, Y_j)$
  else
    $T_{part} = T_{part} \cup (X_j, Y_j)$
  $T = T \setminus (X_j, Y_j)$ ▷ Remove example; maybe it is re-added later
return uncovered examples T, partially covered examples T_{part}, fully covered examples T_{full}

Algorithm 5: Algorithm getCoveredSets for computing the examples, which are partially or fully covered by a given rule [Loza Mencía and Janssen, 2015, Fig. 5]

Based on the data sets T_{full} and T_{part}, which are returned by the subroutine getCoveredSets, the algorithm getReAddSets decides, whether previously removed examples should be re-added to the training data set T, or not. As it can be seen in Algorithm 6, this decision strongly depends on the algorithm's input parameters. If no stopping rules should be used, only partially covered examples are re-inserted into the training process, whereas all fully covered examples are removed. However, it might be desirable to leave the fully covered examples in the training process, since they may be beneficial for the induction of further rules. The possibility of retaining fully covered examples in the training data set is considered, if the percentage of fully covered examples among all covered examples is less than a given parameter τ (usually close to 1). In such case, all partially and fully covered examples are re-added and the recently learned rule is marked as a stopping rule. Instead, if the required percentage is not reached yet, only the partially covered examples are retained in the training process.

Require: Partially and fully covered examples T_{part}, T_{full}, parameter τ
whether using stopping rules, whether re-inserting fully covered examples

$T_{add} = \emptyset$
if use stopping rules **then**
  if full coverage rate $|T_{full}|/(|T_{full}| + |T_{part}|) \geq \tau$ **then** ▷ E.g. 90%
    $T_{add} = \emptyset$ ▷ Do not re-add any example although T_{part}, T_{full} are not empty
  else ▷ Too many partially covered examples
    $T_{add} = T_{part}$ ▷ Re-add partially covered examples
    if re-insert fully covered examples **then**
      $T_{add} = T_{add} \cup T_{full}$ ▷ Re-add also fully covered examples
else
  $T_{add} = T_{part}$ ▷ No stopping rules: re-add partially covered examples
return examples T_{add} to be re-added

Algorithm 6: Algorithm getReAddSet for deciding, whether examples should be re-added to the training data set depending on their covering status, or not [Loza Mencía and Janssen, 2015, Fig. 6]

In order to predict the labels of an unknown test example, the rules, which are contained in a learned multi-label decision list, are applied to the test example successively. Algorithm 7 illustrates how the predicted label vector is constructed by applying the heads of a decision list's rules. The rules are processed in the order of induction. On the one hand, this is necessary with respect to stopping rules, which cause the prediction to be stopped prematurely in order to prevent too many labels from being predicted as relevant. On the other hand, this ensures, that the labels, which are contained in the bodies of label-dependent rules, are already applied to the test example.

Require: Test example X, multi-label decision list R

$\hat{Y} = \langle ?, ..., ? \rangle$
for each rule r in decision list R **do** ▷ Apply rules in the order of induction
  if r covers X **then**
    apply head of r on $\hat{Y}$ if corresponding value in $\hat{Y}$ is unset
    if r is marked as stopping rule or $\hat{Y}$ is complete **then**
      assume all remaining labels in $\hat{Y}$ to be irrelevant
      return prediction $\hat{Y}$
assume all remaining labels in $\hat{Y}$ to be irrelevant
return prediction $\hat{Y}$

Algorithm 7: Algorithm for predicting the label vector of a test example, based on the rules of a multi-label decision list [Loza Mencía and Janssen, 2015, Fig. 7]

According to Algorithm 7 – unless a stopping rule is encountered –, the prediction of a test example's label vector is continued until no rules remain in the decision list or until all labels of the test example are already predicted. Labels, which remain unset after the prediction process is finished, are assumed to be irrelevant.

2.3 Multi-Label Evaluation Metrics

For evaluating the quality of multi-label classifications, different evaluation methods than those that are traditionally used in case of single-class data, are required [Loza Mencía, 2012, Maimon and Rokach, 2005]. For this reason, various measures already known from multi-class classification have been adopted [Loza Mencía, 2012]. In this section, a selection of metrics, which are relevant to the remainder of this work, are discussed. In general, two types of evaluation metrics can be distinguished: i) bipartition and ii) ranking measures [Loza Mencía, 2012, Maimon and Rokach, 2005]. As label-ranking approaches are not part of the present work, the latter are not considered in this section.

2.3.1 Bipartition Evaluation Functions

Bipartition evaluation functions are based on evaluating the differences between true label vectors – also referred to as the *ground truth* (cf. [Maimon and Rokach, 2005]) – and predicted label vectors [Maimon and Rokach, 2005]. Usually, bipartition evaluation metrics can be considered as functions on two-dimensional confusion matrices [Koyejo et al., 2015, Loza Mencía and Janssen, 2015]. The elements of such a confusion matrix represent the *true positives* (*TP*), *false positives* (*FP*), *true negatives* (*TN*) and *false negatives* (*FN*) of a prediction [Koyejo et al., 2015, Loza Mencía, 2012]. Relevant labels are counted as true positives, if they have been set correctly, respectively as false negatives, if they have not been set despite their relevance. Accordingly, irrelevant labels are counted as true negatives, if they have not been set by a prediction, or as false positives, if they have been set mistakenly. Equation 2.5, which is discussed in the following, shows how the elements of a binary confusion matrix are computed in detail. Moreover, in Table 3 the structure of a two-dimensional confusion matrix is illustrated.

C	predicted	not predicted
relevant	TP	FN
irrelevant	FP	TN

Table 3: The structure of a two-dimensional confusion matrix, consisting of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) [Loza Mencía, 2012, Section 2.7.3]

For a given instance X_j and a label λ_i, the elements of an atomic confusion matrix C_i^j are computed as shown in Equation 2.5 below [Loza Mencía, 2012]. The variables y_i and $\hat{y}_i$, which are used in said equation, denote the presence (1) or absence (0) of the label λ_i in the true label vector, respectively in the predicted one.

$$TP_i^j = \begin{cases} 1, & \text{if } y_i = 1 \wedge \hat{y}_i = 1 \\ 0, & \text{otherwise} \end{cases} \qquad FP_i^j = \begin{cases} 1, & \text{if } y_i = 0 \wedge \hat{y}_i = 1 \\ 0, & \text{otherwise} \end{cases}$$
$$TN_i^j = \begin{cases} 1, & \text{if } y_i = 0 \wedge \hat{y}_i = 0 \\ 0, & \text{otherwise} \end{cases} \qquad FN_i^j = \begin{cases} 1, & \text{if } y_i = 1 \wedge \hat{y}_i = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

In the remainder of this work, the number of all relevant and irrelevant labels, which are contained in a confusion matrix, are denoted as P, respectively N. Accordingly, the number of all labels, which are assumed to be relevant by a prediction, is denotated as p, whereas n refers to the number of labels, which are predicted to be irrelevant.

$$P := TP + FN \qquad N := FP + TN$$
$$p := TP + FP \qquad n := TN + FN \tag{2.6}$$

2.3.2 Aggregation and Averaging

When using bipartition functions for evaluating multi-label predictions, which have been made for m instances with n predefined labels being available, $m \cdot n$ atomic confusion matrices can be computed according to the definition of true positives, false positives, true negatives and false negatives given in Equation 2.5 [Loza Mencía, 2012]. However, these individual matrices are not well suited for rating the quality a previously learned model or a single rule. In order to overcome this deficiency, a single score must be calculated by aggregating the confusion matrices, which have been obtained using a bipartition evaluation function δ, according to one of the following strategies: i) *Micro-averaging* is based on aggregating all available information at first and applying the evaluation function afterwards. ii) *Macro-averaging* refers to applying the evaluation function on each available piece of information individually and finally aggregating the results [Loza Mencía, 2012]. When using macro-averaging, an *example-* or *label-based* evaluation is possible. The former is based on averaging evaluations, which have been computed per example. When using the latter, evaluation metrics are calculated for individual labels instead [Maimon and Rokach, 2005]. In favor of formally defining the different averaging strategies in a mathematical way, the following two operators for aggregating, respectively averaging, a sequence of confusion matrices $C_i = (C_1, ..., C_n)$ are declared (cf. [Loza Mencía, 2012]). The $\oplus$ symbol, which is used in the equations below, refers to the cell-wise addition of multiple confusion matrices' elements.

$$\sum_{i=1}^{n} C_i := C_1 \oplus ... \oplus C_n$$
$$\underset{i=1}{\overset{n}{\text{avg}}} C_i := \frac{1}{n} \sum_{i=1}^{n} C_i \tag{2.7}$$

Using the aggregation and averaging operators given in Equation 2.7, the different averaging strategies, which are available for evaluating multi-label predictions, can be defined as follows. The variable i is used to iterate over all available labels $\lambda_1, ...\lambda_n$, whereas the variable j iterates over all of the data set's examples $(X_1, Y_1), ..., (X_m, Y_m)$.

- **(Label and Example-based) Micro-Averaging:** A global confusion matrix is computed by adding up the true positives, false positives, true negatives and false negatives of each example and label. Finally, a single score is calculated by applying the evaluation function δ on the aggregated confusion matrix [Koyejo et al., 2015]. The label-, respectively example-wise, aggregation is commutative, i.e. iterating over the examples at first is computationally equal to first iterating over the labels [Loza Mencía, 2012].

$$\delta\left(\sum_j \sum_i C_i^j\right) \equiv \delta\left(\sum_i \sum_j C_i^j\right) \tag{2.8}$$

- **Example-based (Macro-)Averaging:** At first one confusion matrix is calculated per example by adding up the true positives, false positives, true negatives and false negatives of each label. Applying the evaluation function δ on each obtained confusion matrix afterwards results in m individual values, i.e. one value per example. By calculating the arithmetic mean of these values, the final score is obtained [Koyejo et al., 2015].

$$\operatorname*{avg}_j \delta\left(\sum_i C_i^j\right) \tag{2.9}$$

- **Label-based (Macro-)Averaging:** One confusion matrix is computed per label by aggregating the true positives, false positives, true negatives and false negatives of each example. By applying the evaluation function on each of the obtained confusion matrices, one value is calculated per label. Finally, the arithmetic mean of all of these values is computed in order to obtain the overall score [Koyejo et al., 2015].

$$\operatorname*{avg}_i \delta\left(\sum_j C_i^j\right) \tag{2.10}$$

- **(Label- and Example-based) Macro-Averaging:** At first the evaluation function δ is applied to each atomic confusion matrix, computed as shown in Equation 2.5 for each example and label. Afterwards, the final score is calculated by computing both the label- and example-wise arithmetic mean of all values, which have been obtained this way. Calculating the mean of the obtained values is commutative, i.e. iterating over the examples at first is computationally equal to first iterating over the labels [Loza Mencía, 2012].

$$\operatorname*{avg}_j \operatorname*{avg}_i \delta\left(C_i^j\right) \equiv \operatorname*{avg}_i \operatorname*{avg}_j \delta\left(C_i^j\right) \tag{2.11}$$

When using micro-averaging, examples with a large number of associated labels have a greater influence on the overall performance than those, which are associated with few labels. In contrast, using example-based averaging causes all examples, regardless of their labels, to be weighted equally. Accordingly, when evaluating predictions by using macro-averaging, labels that are associated with many instances have a greater influence than those that only occur sporadically. If each label should be taken into account equally, label-based averaging can be used instead [Loza Mencía, 2012].

In the remainder of this work the following short-hand notation is used in order to specify the averaging technique, which is used together with a particular bipartition evaluation function δ. When using micro-averaging, the evaluation function is referred to as δ_{mm}. Accordingly, the usage of macro-averaging is denotated as δ_{MM} and example- or label-based averaging correspond to the notation δ_{Mm}, respectively δ_{mM}. The two-digit indices, which are used in this notation, specify whether micro- (m) or macro-averaging (M) is used for the example-, respectively label-wise aggregation of confusion matrices. The first symbol refers to the example-wise aggregation, the second one denotes the strategy, which is used for label-wise aggregation, accordingly.

2.3.3 Selected Evaluation Functions

Regardless of the used averaging strategy, choosing an appropriate evaluation function δ for calculating comparable scores is crucial for evaluating multi-label predictions. Such functions are also often referred to as *heuristics* (cf. Janssen [2012]). In this section, various types of evaluation functions are introduced. All of them are further examined in Section 5. Moreover, they are used in order to statistically evaluate the predictive performance of the algorithm, which is proposed in this work, in Section 6. The evaluation functions, which are presented in the following, are defined in terms of true positives, false positives, true negatives and false negatives of a confusion matrix according to the definitions and notations previously given in Section 2.3.1. All of them are surjections $\mathbb{N}^{2x2} \to \mathbb{R}$, mapping the confusion matrix, which characterizes a rule's prediction, to a heuristic value $h \in [0, 1]$. As rules, which reach a higher value, are considered to outperform those with lower values, the used metrics are referred to as *gain metrics*.

- **Precision:** Like most evaluation functions, which are discussed in this section, precision is a well-known metric in the area of binary and multi-class classification [Janssen, 2012, Janssen and Fürnkranz, 2010, Maimon and Rokach, 2005]. Besides that, it can also be used for evaluating multi-label predictions by calculating the percentage of correct predictions among all predicted labels. It is defined according to the following equation [Koyejo et al., 2015, Loza Mencía, 2012]:

$$\delta_{prec} := \frac{TP}{TP + FP} \equiv \frac{TP}{p} \tag{2.12}$$

- **Recall:** This evaluation function is used to measure the percentage of labels, which are assumed to be relevant by a prediction, among all relevant labels according to the ground truth. It is defined as follows [Loza Mencía, 2012]:

$$\delta_{rec} := \frac{TP}{TP + FN} \equiv \frac{TP}{P} \tag{2.13}$$

- **Hamming Accuracy:** This is another evaluation function already known from binary and multi-class classification, where it is traditionally referred to as "Accuracy" [Janssen and Fürnkranz, 2010]. When being used for evaluating multi-label predictions, it calculates the percentage of correctly predicted relevant and irrelevant labels among all labels and is defined according to the equation below:

$$\delta_{hamm} := \frac{TP + TN}{TP + FP + TN + FN} \equiv \frac{TP + TN}{P + N} \tag{2.14}$$

  Hamming accuracy as used in this work is strongly related to the hamming loss metric, which computes the percentage of misclassified labels as shown below. Whereas hamming loss is used to compute an error, hamming accuracy is equal to $1 - \delta_{hammloss}$ and therefore is the corresponding gain metric [Koyejo et al., 2015]. As opposed to gain metrics, which must be maximized by a classifier in order to achieve a better predictive performance, *loss metrics* must be minimized.

$$\delta_{hammloss} := \frac{FP + FN}{TP + FP + TN + FN} \equiv \frac{FP + FN}{P + N} \tag{2.15}$$

  In the present work, hamming accuracy is preferred over hamming loss, because of its accordance with all other used performance measures, which are gain metrics as well.

- **F-Measure:** As shown in Equation 2.16 below, the F-Measure is defined as the harmonic mean of precision and recall. The parameter $\beta \in [0, \infty]$ allows to trade off between precision and recall. If $\beta < 1$, precision has a greater impact on the measured performance. If $\beta > 1$, the metric becomes more recall-oriented instead [Chinchor, 1992, Sasaki, 2007].

$$\delta_F := \frac{(\beta^2 + 1) \cdot \delta_{prec} \cdot \delta_{rec}}{\beta^2 \cdot \delta_{prec} + \delta_{rec}} = \frac{\beta^2 + 1}{\frac{\beta^2}{\delta_{rec}} + \frac{1}{\delta_{prec}}} \tag{2.16}$$

A commonly used variant of the F-Measure is the one that weights precision and recall equally by setting $\beta = 1$. It is often referred to as the "F1-Measure" and calculates as follows [Sasaki, 2007]:

$$\delta_{F1} := \frac{2 \cdot \delta_{prec} \cdot \delta_{rec}}{\delta_{prec} + \delta_{rec}} = \frac{2}{\frac{1}{\delta_{rec}} + \frac{1}{\delta_{prec}}} \tag{2.17}$$

- **Subset Accuracy:** This metric is slightly different from all other evaluation functions, which are mentioned in this section, as it is only defined in terms of example-based averaging. It is based on comparing the predicted and true label vectors $\hat{Y}_j$ and Y_j of each example X_j. If the label attributes of both vectors do exactly match, the performance for an individual example evaluates to 1. If the vectors differ in at least one label attribute, the performance is measured as 0 instead [Loza Mencía, 2012, Loza Mencía and Janssen, 2015, Zhu et al., 2005]. When calculating the mean of the performances, which have been obtained for all m examples of a data set, as denoted by Equation 2.18 below, subset accuracy corresponds to the percentage of perfectly predicted label vectors among all examples [Loza Mencía, 2012].

$$\delta_{acc} := \frac{1}{m} \sum_{j=1}^{m} [Y_j = \hat{Y}_j] \text{ , with } [x] = \begin{cases} 1, & \text{if } x \text{ is true} \\ 0, & \text{otherwise} \end{cases} \tag{2.18}$$

The notation, which uses square brackets for denoting the conversion of boolean expressions into integer values 0 or 1, as it can be seen in Equation 2.18 above, is known as "Iverson bracket notation" [Knuth, 1992]. It is used various times throughout the remainder of the present work.

3 Searching through the Label Space for finding Multi-Label Heads

In order to induce multi-label head rules, the algorithm, which is proposed in this work, relies on finding the best multi-label head for a given rule's body. However, due to the exponential complexity of an exhaustive search through the label space $\mathbb{L} = (\lambda_1, ..., \lambda_n)$, which requires 2^n steps [Loza Mencía, 2012], it is not feasible to evaluate all possible heads, if many labels are available. For this reason, a way for pruning the search by leaving out the evaluation of unpromising label combinations is required. The algorithm, which is proposed in this work, is able to perform pruned searches through the label space based on two different properties of multi-label evaluation metrics – namely *anti-monotonicity* and *decomposability*. Prior to the presentation of the algorithm, which is given in Chapter 4, both of these properties are formally defined in this chapter. Whereas anti-monotonicity is discussed in Section 3.2, Section 3.3 focuses on decomposable evaluation metrics. In addition to the formal definitions, which are given in this chapter, multiple examples, that illustrate the search for multi-label heads using different pruning strategies, are given as well. Besides emphasizing the need for pruned searches, their purpose is to demonstrate how the performances of multi-label head rules are measured. In this work, it is distinguished, whether a *rule-independent* or *rule-dependent* evaluation strategy is used. Both strategies are discussed in Section 3.1 below.

3.1 Rule-dependent vs. Rule-independent Evaluation

As already mentioned, the evaluation of a multi-label head rule depends on whether the rule-dependent or rule-independent evaluation strategy should be used. This differentiation corresponds to the labels, which are taken into account for measuring the rule's performance. The individual strategies are defined as follows:

- **Rule-independent:** All labels $\lambda_1, ..., \lambda_n \in \mathbb{L}$ are taken into account for measuring a rule's performance, regardless of the rule's head. If the head does not contain a label attribute $\hat{y}_i$, corresponding to a certain label λ_i, this is handled as if a label attribute $\hat{y}_i = 0$ would be included, i.e. as if the label would be predicted as irrelevant.
- **Rule-dependent:** Only if a rule does contain a label attribute $\hat{y}_i$ in its head, the corresponding label λ_i is taken into account for measuring the rule's performance.

When using the rule-independent evaluation strategy, all labels are taken into account for measuring the performance of a rule, regardless of whether they are included in the rule's head, or not. Labels, which are not included, are assumed to be predicted as irrelevant. This assumption is based on how a test example's label vector is constructed during the prediction process by successively applying the rules, which are contained by a previously learned multi-label decision list. The prediction algorithm (which is discussed in detail in Section 4.4) considers labels, that remain unset after all rules of the decision list have been processed or after a stopping rule has been encountered, to be irrelevant. Using the rule-independent evaluation strategy is expected to introduce a bias towards learning fully set heads, which predict all available labels. However, when taking into consideration, that separate-and-conquer algorithms aim at learning several rules, utilizing the rule-dependent evaluation strategy might be more appropriate for evaluating an individual rule's performance. This is, because when predicting the labels of a test example, usually multiple rules contribute to the predicted label vector, each one of them predicting the absence or presence of only a few labels. When only taking the labels into account, which are actually set by a particular rule, the performance of that particular rule exclusively depends on these labels, instead of being affected by labels, which are out of the rule's scope and might be predicted by other rules instead. Both evaluation strategies are examined in terms of anti-monotonicity and decomposability in Chapter 5. Furthermore, the influence of using either the rule-dependent or rule-independent evaluation strategy on the proposed algorithm's performance is discussed in Chapter 6. In the remainder of this work, the symbols $\perp\!\!\!\perp$ and $\not\perp\!\!\!\perp$ are used as a shorthand notation in order to indicate, that the rule-independent, respectively rule-dependent, evaluation strategy is used.

3.2 Anti-Monotonicity

In Definition 3.1, the anti-monotonicity property is formally defined. If the definition holds for a specific evaluation function, using a specific averaging and evaluation strategy, this basically guarantees, that when adding a label attribute to a multi-label head rule's head causes the rule's performance to decrease, the heads, which result from adding additional attributes, cannot reach the best possible performance anymore. As a result, it is possible to leave out the evaluation of these unpromising heads without risking, that the best possible head is not found. In addition, the applicability of pruning based on anti-monotonicity is further restricted by requiring the properties of *monotonicity*, as given in Definition 3.2, to not be met. The motivation behind this additional restriction is discussed below.

Definition 3.1 - Anti-Monotonicity: *Let $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ denote two multi-label head rules consisting of a common body B and head $\hat{Y}_p$, respectively $\hat{Y}_s$. Both heads contain label attributes $\hat{y}_i \in G$ with $G = \{1\}$ or $G = \{0, 1\}$. It is further assumed, that the subset relationship $\hat{Y}_p \subset \hat{Y}_s$ holds and therefore the head $\hat{Y}_s$ contains additional label attributes beyond those of $\hat{Y}_p$. Given a particular averaging strategy and using either the rule-dependent or rule-independent evaluation strategy, an evaluation function δ is considered to be anti-monotonous, if the following conditions are met:*

i) When adding a label attribute to a multi-label head rule's head causes the rule's performance on a data set T to decrease, by adding additional attributes the best possible performance h_{max} cannot be reached anymore:

$$\hat{Y}_p \subset \hat{Y}_s \wedge \delta(\hat{Y}_s \leftarrow B, T) < \delta(\hat{Y}_p \leftarrow B, T) \Longrightarrow \delta(\hat{Y}_a \leftarrow B, T) < h_{max}\ ,\ \forall \hat{Y}_a (\hat{Y}_s \subset \hat{Y}_a)$$

ii) Regarding the given averaging and evaluation strategy, the evaluation function δ must not be monotonous according to Definition 3.2.

In order to illustrate, how searches through the label space, which aim at finding the best multi-label head for a given rule's body, can be pruned by exploiting the properties of anti-monotonous evaluation functions, an example is given in the following. By showing all evaluations, which must be performed by an exhaustive search, as well as by highlighting the evaluations, which can be left out by a pruned search, the need to reduce the computational complexity is emphasized.

Example 3.1 - Pruning Searches Through the Label Space based on Anti-Monotonicity: In Table 4, the label vectors of fictional training examples are given. One-half of them is assumed to be covered by a given body, the other half is not. The aim of a search through the label space, as it is discussed in this example, is to find the multi- or single-label head rule, which models the labels of the covered examples best. In order to be able to compare the possible rules with each other, a heuristic value is calculated per rule by using a specific evaluation function, averaging strategy and evaluation strategy.

		λ_1	λ_2	λ_3	λ_4
Not covered	Y_1	0	1	1	1
	Y_2	1	1	1	1
	Y_3	0	1	0	0
Covered	Y_4	0	1	1	0
	Y_5	1	1	0	0
	Y_6	1	1	0	0

Table 4: Exemplary label vectors of training examples used by Figure 1, as well as by Figure 2, and given the label space $\mathbb{L} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. Some of the examples are assumed to be covered by a given rule's body, some are not.

The search for finding the best head for a given rule's body can be visualized as a search tree similar to the one, which is shown in Figure 1 below. The nodes of the tree correspond to the evaluation of label combinations. The root node represents an empty head with undefined performance. The edges of such tree correspond to adding an additional label attribute to the head, which is represented by the preceding node. Because equivalent heads are prevented from being evaluated multiple time, the search tree is unbalanced. Whereas many label combinations are evaluated in the most-left branch of the tree, the number of evaluations decreases from left to right. In the present example micro-averaged hamming accuracy – as previously defined in Equation 2.8, respectively Equation 2.14 –, together with the rule-independent evaluation strategy (cf. Section 3.1), is used for measuring the heuristic values h of possible single- and multi-label head rules. When using an exhaustive search, all possible label combinations must be evaluated. The resulting tree is independent of whether a breadth-first or depth-first search is used. In Figure 1, increases of the measured performance, which result from adding an additional label attribute to the rule's head, are indicated using green arrows (→). Decreases of the performance are indicated by using red arrows (→) accordingly. If adding a label attribute does not have an impact on the rule's performance at all, black arrows (→) are used. Note, that in the present example only label attributes, which predict the presence of labels (i.e. attributes of the form $\hat{y}_i = 1$) are considered. This corresponds to the target $G = \{1\}$. In general, by using the targets $G = \{0, 1\}$, multi-label heads can also predict the absence of labels (using attributes of the form $\hat{y}_i = 0$). This requires both possible forms of a particular label attribute to be evaluated in each node of the search tree. The variant, which reaches a higher performance according to the used evaluation function, is finally added to the rule's head and propagated to the child nodes.

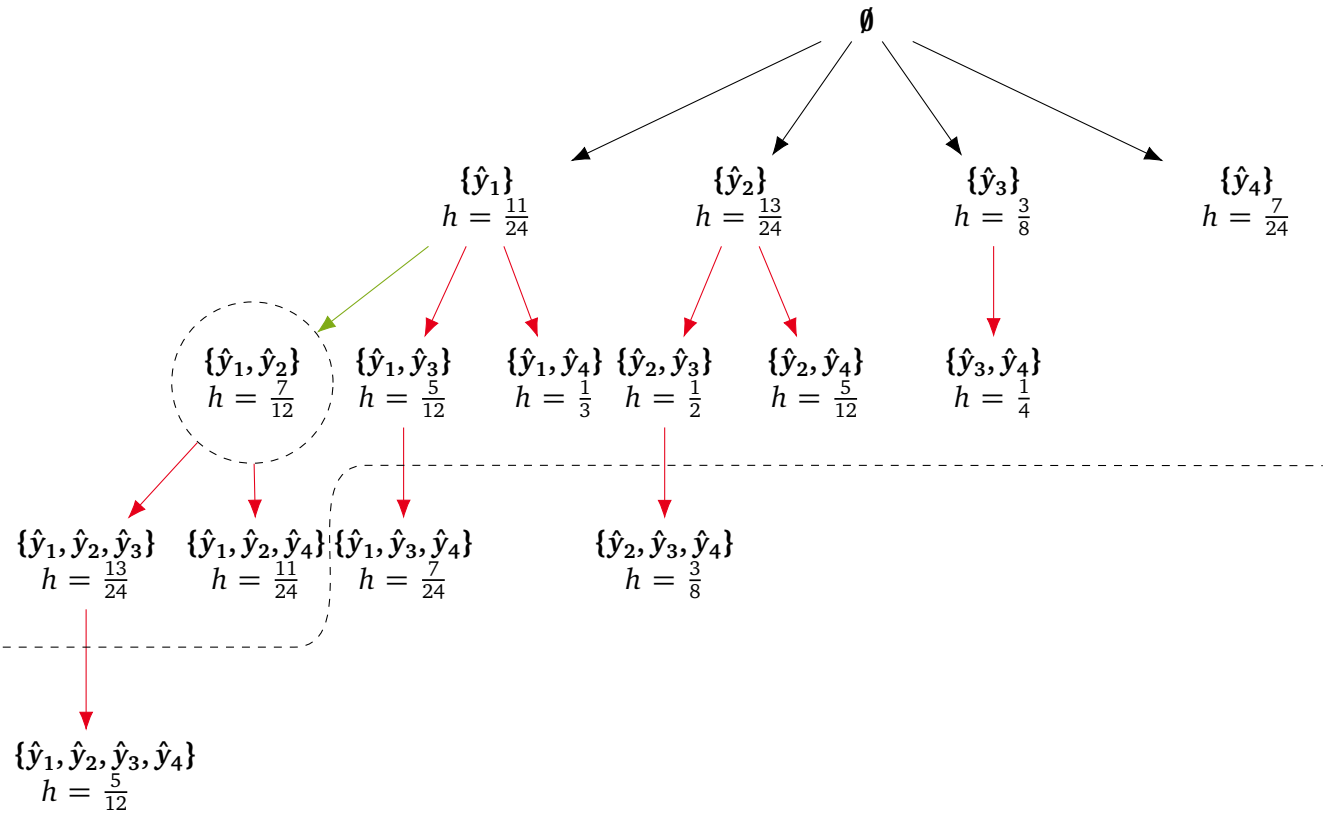

Figure 1: Search through the label space for finding the best multi-label rule head given the examples in Table 4 and using micro-averaged hamming accuracy, together with the rule-independent evaluation strategy, for performance evaluation. The dashed line (- - -) indicates the label combinations, which must not be considered, when pruning the search according to the anti-monotonicity property, which is given in Definition 3.1.

According to the rule-independent evaluation strategy, which is used in the present example, all labels – regardless of whether they are included in a rule's head, or not – are taken into account for calculating a rule's performance h. Furthermore, as the head is only applied to examples, which are covered by the rule's body, covered examples are treated differently than uncovered examples. On the one hand, relevant labels of uncovered examples are counted as false negatives and irrelevant labels contribute to the true negatives. On the other hand, the labels of covered examples are counted as true positives, if they are predicted correctly, respectively as false positives, if they are predicted incorrectly. This contradicts the definition, which is given in Equation 2.5, and aims at handling predicted labels equally, regardless of whether they are relevant or irrelevant. The need for using the divergent definition is discussed more detailed in Section 4.3, when discussing the proposed algorithm. For a better understanding of how the performances of multi-label head rules are calculated, an exemplary calculation is shown below. It illustrates, how the performance of the rule with the head $\{\hat{y}_1, \hat{y}_3\}$ is calculated. According to the notation, which is shown in Table 1, the index j is used for iterating over the data set's examples, whereas the index i iterates over the available labels. The colors, which are used in the equation for highlighting true positives, false positives, true negatives and false negatives, correspond to the color highlighting used in Table 4.

$$h_{\hat{y}_1,\hat{y}_3} = \frac{\sum_i \sum_j TP_i^j + \sum_i \sum_j TN_i^j}{\sum_i \sum_j TP_i^j + \sum_i \sum_j FP_i^j + \sum_i \sum_j TN_i^j + \sum_i \sum_j FN_i^j} = \frac{6+4}{6+6+4+8} = \frac{5}{12}$$

According to Figure 1, the label combination $\{\hat{y}_1, \hat{y}_2\}$, which reaches a performance of $\frac{7}{12}$, is considered to be the best choice. In order to induce multi-label head rules, which predict as many labels as possible, heads that consist of more label attributes should be preferred over those with fewer attributes. The dashed line (- - -) in Figure 1 indicates the label combinations, which must not be considered when pruning the search under the assumption, that the hamming accuracy metric fulfills the properties of anti-monotonicity when used together with micro-averaging and the rule-dependent evaluation strategy (in Section 5.2.3 this assumption is proved to hold). As it can be seen, the highest rated label combination $\{\hat{y}_1, \hat{y}_2\}$ is still discovered by a pruned search, despite the reduced search complexity.

Definition 3.2 - Monotonicity: *Let $\hat{Y}_p \leftarrow B$ denote a multi-label head rule with body B and head $\hat{Y}_p$. The head consists of an arbitrary number of label attributes using the targets $G = \{1\}$ or $G = \{0, 1\}$. Given a particular averaging strategy and using either the rule-dependent or rule-independent evaluation strategy, an evaluation function δ is considered to be monotonous, if adding an additional label attribute, which is not already contained by the head $\hat{Y}_p$, never causes the rule's performance on the data set T to decrease:*

$$\delta(\hat{Y}_a \leftarrow B, T) \geq \delta(\hat{Y}_p \leftarrow B, T)\,,\ \forall \hat{Y}_a \left(\hat{Y}_p \subset \hat{Y}_a\right)$$

According to Definition 3.1, anti-monotonous evaluation functions must not be monotonous. This is, because otherwise, adding label attributes to a multi-label head would never cause the performance of the corresponding rule to decrease. This would prevent any evaluations from being pruned, resulting in a computational complexity, which is equal to that of an exhaustive search. Moreover, all rules, which are induced using a monotonous evaluation function, would predict all available labels to be either relevant, or irrelevant, which results in a bad predictive performance. An example of a monotonous evaluation function is shown in the following example.

Example 3.2 - Monotonous Evaluation Functions: In Figure 2, a search through the label space $\mathbb{L} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ is illustrated. The exemplary label vectors, which are given in Table 4, are reused in the present example. For measuring the performances of possible multi-label head rules, the recall metric, which is given in Equation 2.13, is used together with example-based averaging as defined in Equation 2.9. Furthermore, the rule-dependent evaluation strategy, as discussed in Section 3.1, is

used in this example. Because the used evaluation function is monotonous (which is proved in Section 5.1.2.3), adding an additional label attribute to a multi-label head does never cause the measured performance to decrease. Instead, the performance remains the same, which is indicated by using black arrows (→) in Figure 2, or even increases as indicated by the green arrows (→). As a result, all potential multi-label heads must be evaluated, resulting in a computational complexity of 2^n. This corresponds to the complexity of an exhaustive search. As the algorithm, which is proposed in this work, prefers multi-label heads, which consist of more label attributes, over those that consist of fewer attributes, the head $\{\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4\}$, which predicts all available labels as relevant, is considered to be the best choice. However, when taking a look at the label vectors of the used training examples, it becomes obvious, that predicting some labels to be irrelevant would be a better a choice. E.g. label λ_4 is not associated with any of the covered examples and therefore it should be predicted as irrelevant.

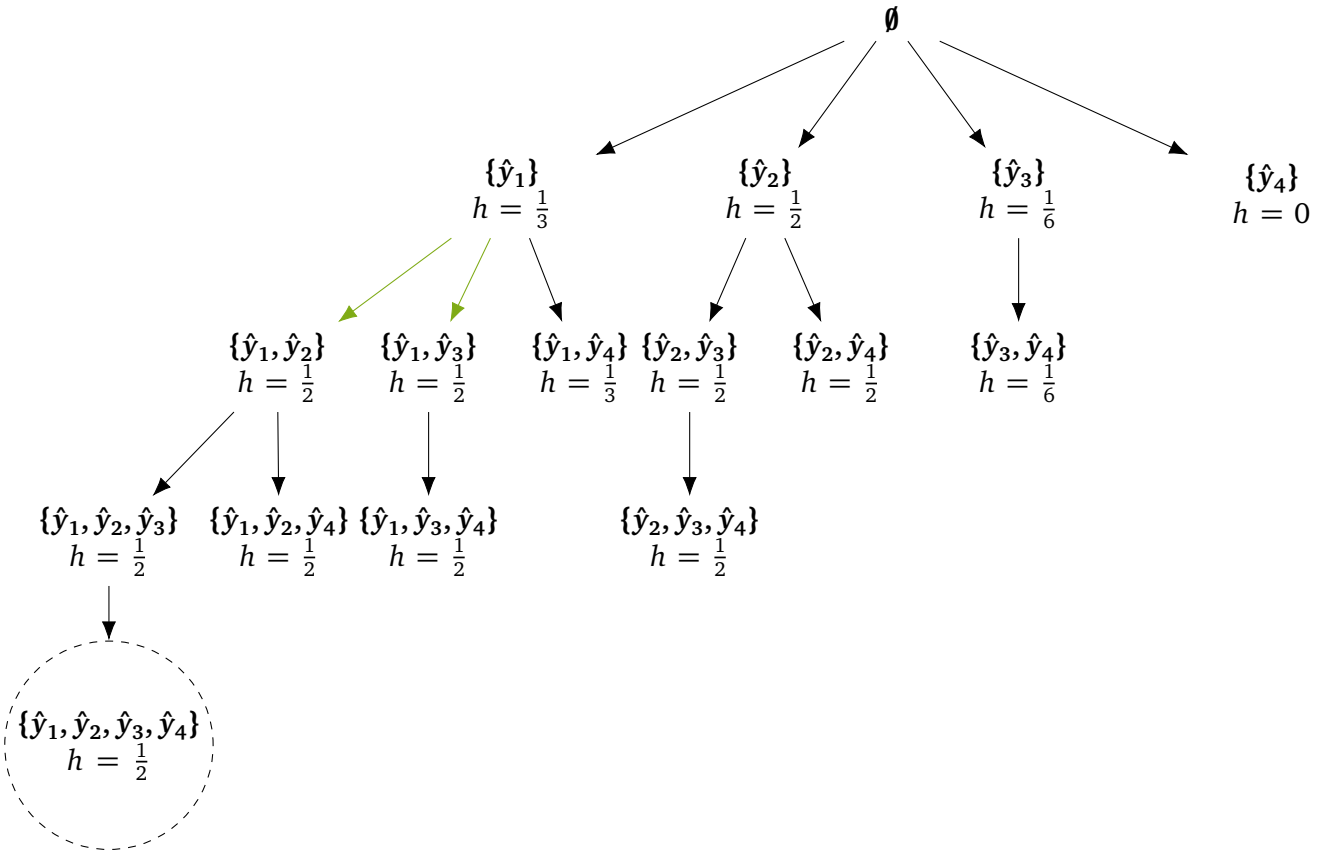

Figure 2: Search through the label space for finding the best multi-label rule head given the examples in Table 4 and using example-based recall, together with the rule-dependent evaluation strategy, for performance evaluation.

3.3 Decomposable Evaluation Metrics

In addition to the anti-monotonicity property, which is specified in Definition 3.1 of the previous section, another property of evaluation functions, referred to as *decomposability*, is exploited in this work for pruning searches through the label space as well. When using a so-called *decomposable evaluation metric*, no deep searches through the label space must be performed in order to find the best multi-label head for a given rule's body. Instead, given a label space $\mathbb{L} = (\lambda_1, ...\lambda_n)$, the best multi-label head can be deduced from the performance measurements, which are obtained by considering each of the n labels individually. Definition 3.3, which is given below, formally defines the properties of decomposable evaluation functions.

Definition 3.3 - Decomposability: *Let $\hat{Y} \leftarrow B$ denote a multi-label head rule with body B and an arbitrary head $\hat{Y}$. Given a particular averaging strategy and using either the rule-dependent or rule-independent evaluation strategy, an evaluation function δ is considered to be decomposable, if the following conditions are met:*

i) If the multi-label head rule $\hat{Y} \leftarrow B$ contains a label attribute $\hat{y}_i \in \hat{Y}$ for which the corresponding single-label head rule $\hat{y}_i \leftarrow B$ does not reach the best possible performance h_{max} on the data set T, the multi-label head rule cannot reach that performance either. The other way around, if a multi-label head rule does not reach the performance h_{max}, at least one of the corresponding single-label head rules does not reach that performance either.

$$\exists i \left(\hat{y}_i \in \hat{Y} \wedge \delta(\hat{y}_i \leftarrow B, T) < h_{max} \right) \Longleftrightarrow \delta(\hat{Y} \leftarrow B, T) < h_{max}$$

ii) If all single-label head rules $\hat{y}_i \leftarrow B$, which correspond to the label attributes, which are contained by a multi-label head $\hat{Y}$, reach the best possible performance h_{max} on the data set T, the multi-label head rule $\hat{Y} \leftarrow B$ reaches that performance as well. The other way around, if a multi-label head rule reaches the performance h_{max}, all corresponding single-label head rules also reach that performance.

$$\delta(\hat{y}_i \leftarrow B, T) = h_{max} \; , \; \forall \hat{y}_i \left(\hat{y}_i \in \hat{Y} \right) \Longleftrightarrow \delta(\hat{Y} \leftarrow B, T) = h_{max}$$

iii) Regarding the given averaging and evaluation strategy, the evaluation function δ must not be monotonous according to Definition 3.2.

The properties of decomposability, according to Definition 3.3, are more restrictive than those of anti-monotonicity, which are given in Definition 3.1. As a result, if the definition of decomposability is met by an evaluation function, the definition of anti-monotonicity is implied to be met by said evaluation function as well. This kind of implicational relationship is expressed by Lemma 3.1, which is shown below.

Lemma 3.1: *If an evaluation function δ is decomposable according to Definition 3.3, given a specific averaging and evaluation strategy, this implies that it is also anti-monotonous according to Definition 3.1.*

Proof: On the one hand, if the property, which is given in Definition 3.3 iii), is met, this implies the corresponding property in Definition 3.1 ii) to be met as well. On the other hand, if the properties, which are given in Definition 3.3 i) and ii), are fulfilled, it follows, that the property, which is given in Definition 3.1 i), holds as well due to the following equation:

$$\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta(\hat{Y}_s \leftarrow B, T) < \delta(\hat{Y}_p \leftarrow B, T) \\
\Longrightarrow &\delta(\hat{Y}_s \leftarrow B, T) < h_{max} \\
\xRightarrow[\text{Definition 3.3 i)}]{\text{w.r.t.}} &\exists \left(\hat{y}_i \in \hat{Y}_s \wedge \delta(\hat{y}_i \leftarrow B, T) < h_{max} \right) \\
\Longrightarrow &\exists \left(\hat{y}_i \in \hat{Y}_a \wedge \delta(\hat{y}_i \leftarrow B, T) < h_{max} \right) \; , \; \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
\Longrightarrow &\delta(\hat{Y}_a \leftarrow B, T) < h_{max} \; , \; \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right)
\end{aligned} \tag{3.1}$$

■

As a result of Lemma 3.1, if an evaluation function is proved to be decomposable according to Definition 3.3, it can also be considered to be anti-monotonous according to Definition 3.1. Consequently, when

using such an evaluation strategy, searches through the label space can be pruned according to the properties of anti-monotonicity as previously demonstrated in Example 3.1. However, if the properties of decomposability are met, it is possible to prune searches even more extensively. This is illustrated by the following example.

Example 3.3 - Pruning Searches through the Label Space based on Decomposability: In this example, the rule-dependent evaluation strategy is utilized in order to measure the performances of multi-label head rules using micro-averaged precision. It is further assumed, that the used evaluation function is decomposable according to Definition 3.3 (this assumption is proved to be true in Section 5.1.1.1). The example is based on the training examples, which are shown in Table 5. Some of the examples are assumed to be covered by a given rule's body and some are not.

		λ_1	λ_2	λ_3	λ_4
Not covered	Y_1	0	1	1	0
	Y_2	1	1	1	1
	Y_3	0	0	1	0
Covered	Y_4	0	1	1	0
	Y_5	1	1	0	0
	Y_6	1	0	0	0

Table 5: Exemplary label vectors of training examples used by Figure 3 and given the label space $\mathbb{L} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. Some examples are covered by a given rule's body, some are not.

Figure 3 shows the search tree, which results from an exhaustive search for the best multi-label head, given the exemplary label vectors, which are shown in Table 5.

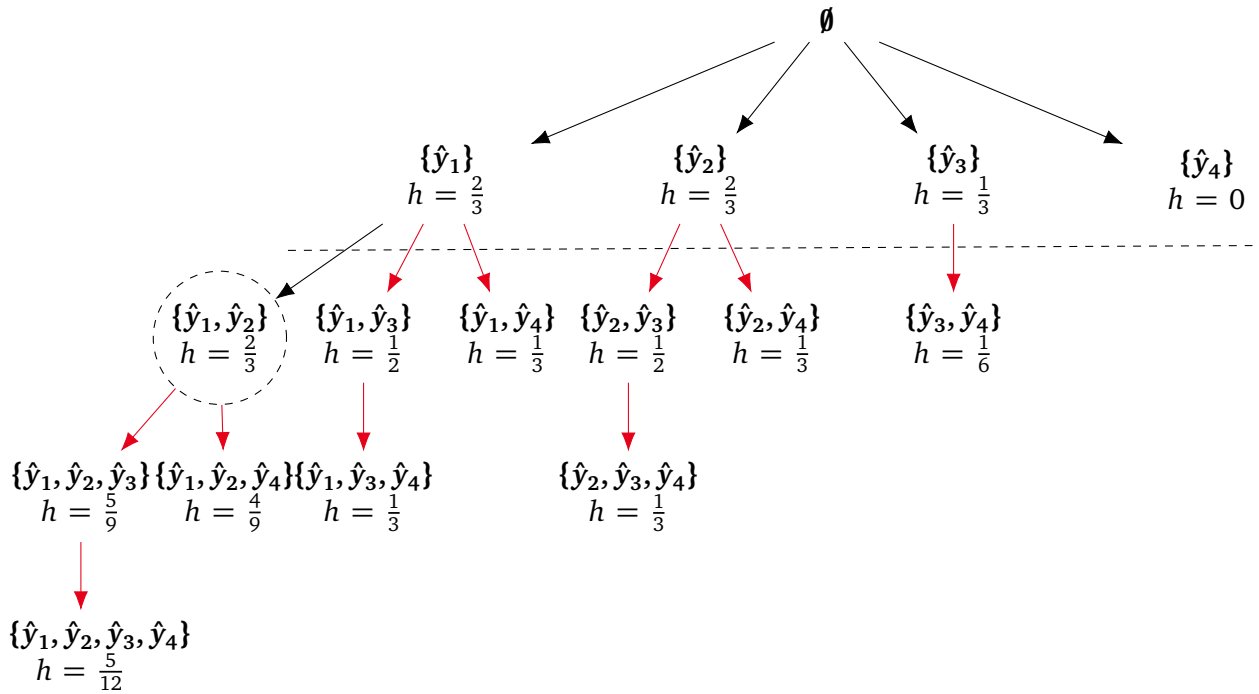

Figure 3: Search through the label space for finding the best multi-label rule head given the examples in Table 5 and using micro-averaged precision, together with the rule-dependent evaluation strategy, for performance evaluation. The dashed line (- - -) indicates the label combinations, which must not be evaluated according to the properties of decomposable evaluation functions, which are given in Definition 3.3.

In Figure 3 – similar to the earlier examples in this chapter –, decreases in performance, which result from adding additional label vectors to a head, are indicated in said figure by using red arrows (→). Increases are indicated by using green arrows (→) accordingly. Black arrows (→) are used, when adding a label attribute does not affect the performance. According to Figure 3, the multi-label head $\{\hat{y}_1, \hat{y}_2\}$, which reaches a performance of $\frac{2}{3}$, is considered to be the best choice. When only taking the single-label heads $\{\hat{y}_1\}$, $\{\hat{y}_2\}$, $\{\hat{y}_3\}$ and $\{\hat{y}_4\}$ into account, the heads $\{\hat{y}_1\}$ and $\{\hat{y}_2\}$, which both reach the best possible performance of $\frac{2}{3}$, outperform the remaining single-label heads. According to the properties of decomposable evaluation metrics, this allows to conclude, that the best multi-label head contains both of these label attributes in its head. This is, because including at least one of the remaining label attributes in the head causes the performance of the resulting multi-label head rule to be lower than the best possible performance. As a result, it is possible to determine the best multi-label head without the need to explicitly evaluate its performance.

4 An Algorithm for Learning Multi-Label Head Rules

In this chapter, the algorithm, which is proposed in the work at hand, is presented. As already mentioned, it is based on the separate-and-conquer algorithm for learning single-label head rules, which has recently been proposed by Loza Mencía and Janssen [2015]. That underlying algorithm is discussed in detail in Section 2.2.3. As the original algorithm, as well as the variant, which is presented at this point, are both based on the paradigms of separate-and-conquer rule learning algorithms (cf. Section 2.2.1), they share a common structure. The subroutines GETCOVEREDSETS (cf. Algorithm 5) and GETREADDSETS (cf. Algorithm 6) are identical for both approaches and therefore they are not discussed here. According to Algorithm 3, which illustrates the basic structure of both separate-and-conquer algorithms, a multi-label decision list is learned by successively inducing rules. Whenever a new rule is induced, the training examples it covers are eventually removed from the training data set, depending on the fraction of labels, which are predicted by already learned rules. Despite these similarities, both approaches significantly differ in how individual rules are learned. Whereas the original algorithm is based on inducing single-label head rules for each label individually and finally choosing the best among them (cf. Algorithm 4), for learning multi-label head rules a different approach is required. As such rules may contain several labels in their heads, the available labels can not be handled in an isolated manner. Instead of finding the best body for given labels, the algorithm for inducing multi-head rules is based on finding the best head for potential bodies. The alternative implementation of the subroutine FINDBESTGLOBALRULE, which is used by said approach, is shown in Algorithm 8 below.

Require: Original training data set T, current training data set $T_{current}$
targets G (either $G = \{1\}$ or $G = \{0, 1\}$), evaluation function δ,
whether to use rule-dependent or rule-independent evaluation strategy,
the averaging strategy to use

```
1  r_best = ∅ ← ∅
2  r_best.h = −∞                          ▷ Initialize performance of new rule
3  improved = true
4  while improved do                      ▷ Until no improvements possible
5     r = REFINERULE(T, T_current, r_best, G, δ)   ▷ Refine rule by adding a condition and updating its head
6     if r.h > r_best.h then
7        r_best = r                       ▷ Replace by better rule
8     else
9        improved = false
10 return best rule r_best
```

Algorithm 8: Algorithm FINDBESTGLOBALRULE for inducing a new multi-label head rule, based on the current training data set

The rule induction process, which is shown in Algorithm 8 above, corresponds to a top-down search, starting with the most generic rule – i.e. a rule with no conditions in its body –, which covers all of the training data set's examples. The rule is then iteratively refined by successively adding additional conditions to its body and choosing a suitable multi-label head each time a condition is added. Adding conditions to a rule's body causes the rule to become more specific, i.e. a subset of the originally covered training examples is covered by the refined rule. This requires the head of the rule to be updated as well, because predicting different labels might result in a higher predictive performance with respect to the changes in covered examples. According to Algorithm 8, a rule is refined until no improvement in terms of the measured performance can be achieved anymore. When this termination criterion is met, the refined rule is returned and added to the decision list.

4.1 Refinement of Rule Conditions

As it can be seen in Algorithm 8, for refining a rule by adding additional conditions to its body, the subroutine REFINERULE is used. It is responsible for determining the best possible refinement of a rule, as well as for finding a corresponding multi-label head. For this reason, each possible combination of refinement and corresponding multi-label head must be taken into consideration. Finally, the best rule among all possible refinements is chosen according to a performance evaluation on the given training data set. The structure of the subroutine REFINERULE, as used by proposed algorithm, is shown in Algorithm 9 below.

Require: Original training data set T, current training data set $T_{current}$,
rule r, targets G, evaluation function δ,
whether to use rule-dependent or rule-independent evaluation strategy,
the averaging strategy to use

1 $r_{best} = r$
2 **for each** possible condition $c \in$ GETATTRIBUTECONDITIONS$(T) \cup$ GETLABELCONDITIONS$(T_{current})$ **do**
3 **if** $c \notin r.body$ **then**
4 $r_{refined} = r$
5 $r_{refined}.body \cup c$ ▷ Add condition to rule's body
6 $r_{refined} =$ FINDBESTHEAD$(T, T_{current}, r_{refined}, G, \delta)$
7 **if** $r_{refined}.h > r_{best}.h$ **then**
8 $r_{best} = r_{refined}$ ▷ Replace by refined rule
9 **return** best refined rule r_{best}

Algorithm 9: Algorithm REFINERULE for refining the body of a multi-label head rule by adding additional conditions. Each refinement requires to update the rule's head as well

The conditions, which are available for being added to a rule's body in order to refine it, result from the attributes and labels of the given training data set. Whereas attribute conditions are tests on the training examples' values, label conditions allow to check, whether particular labels are associated with examples, or not. This enables to induce label-dependent rules (cf. Table 2, which are able to model correlations between labels as previously discussed in Section 2.1.2. In Algorithm 9, the possible attribute and label conditions are retrieved using the subroutine GETATTRIBUTECONDITIONS, respectively GETLABELCONDITIONS. The operation of both of these subroutines is discussed in detail in the following two subsections.

4.1.1 Attribute Conditions

Algorithm 10 illustrates the operation of the subroutine GETATTRIBUTECONDITIONS, which can be used to retrieve all possible attribute conditions for refining a rule's body. The algorithm takes all available attributes of the given training data set T into consideration. The values, which are available for individual conditions, depend on whether the respective attribute is nominal or numeric. In case of a nominal attribute, a condition is created for each of the attribute's values (e.g. the values `true` and `false` if it is a boolean attribute). If the attribute is numeric instead, the conditions depend on the examples, which are present in the given training data set. This requires the data set's examples to be sorted by the respective attribute in ascending order, beforehand. Afterwards, for each pair of neighboring examples, a *split point* is calculated (cf. Algorithm 10, line 10). For each split point, two conditions – using the $\leq$, respectively the $\geq$ operator, for comparing an example's value to a split point – are created (cf. Algorithm 10, line 11). Note, that it is not necessary to create conditions for split points between two neighboring examples, which are associated with an identical set of labels (cf. Algorithm 10, line 9). This is, because the conditions of a rule are supposed to discriminate between examples for which the rule's prediction is correct and those for which the prediction is incorrect. However, when choosing such split point as a condition's value, the condition mistakenly discriminates between examples for which the same prediction is correct.

Require: Training data set T

1 $C = \emptyset$ ▷ Start with empty set of conditions
2 **for each** attribute A_k **do**
3 **if** A_k is nominal **then** ▷ Attribute is nominal
4 **for each** value v of A_k **do** ▷ Add condition for each nominal value
5 $C = C \cup (A_k = v)$
6 **else** ▷ Attribute is numeric
7 sort examples $(X_j, Y_j) \in T$ by values $v_k \in X_j$ in increasing order
8 **for each** example $(X_j, Y_j) \in T$ with $j > 1 \wedge j < |T|$ **do**
9 **if** $Y_{j-1} \neq Y_j$ **then** ▷ Only consider neighbours with different label vectors
10 $v = v_{k-1} + (v_k - v_{k-1})/2$ with $v_{k-1}, v_k \in X_j$ ▷ Calculate split point between neighbours
11 $C = C \cup (A_k \geq v) \cup (A_k \leq v)$ ▷ Add two conditions for each split point
12 **return** attribute conditions C

Algorithm 10: Algorithm getAttributeConditions for retrieving all possible conditions, which are based on the training data set's attributes

4.1.2 Label Conditions

In order to be able to induce label-dependent rules, the labels, which are predicted by already learned rules, can be used as the conditions of new rules. Algorithm 11 illustrates how the label conditions, which are available at a particular point in the rule induction process, can be retrieved using the subroutine getLabelConditions. The labels, which are available for creating conditions, are taken from the current training data set $T_{current}$. Whenever a new rule is learned by the proposed algorithm, its head is applied to the covered training examples (cf. Algorithm 5, line 3), which enables to keep track of already predicted labels. For each of the available labels, two conditions – for testing the presence, respectively the absence of the respective label – are created (cf. Algorithm 11, line 4).

Require: Current training data set $T_{current}$

1 $C = \emptyset$ ▷ Start with empty set of conditions
2 **for each** example $(X_j, Y_j) \in T_{current}$ **do**
3 **for each** label attribute $y_i \in Y_j$ with $y_i \neq ?$ **do**
4 $C = C \cup (y_i = 0) \cup (y_i = 1)$ ▷ Add two conditions for each already learned prediction
5 **return** label conditions C

Algorithm 11: Algorithm getLabelConditions for retrieving all possible conditions, which are based on already predicted labels

The reason why only labels, which are already predicted by previously induced rules, are taken into account, lies in how the decision list, which is learned by the proposed algorithm, is processed in order to predict the labels of an unknown test example. During the prediction process, the rules, which are contained by the decision list, are applied to the test example successively. Initially, all labels of the given example are unset. Only if a rule covers the example, the labels, which correspond to the label attributes in the rule's head, are set. This causes the number of set labels to increase as the decision list is processed. If a rule would depend on a label, which is not set by any of its predecessors, there would be no chance of the label being set once the rule is processed. As a result, the rule would not cover the given example, rendering it useless. The application of a learned decision list for predicting the labels of unknown examples is discussed in a more detailed manner in Section 4.4.

4.2 Finding Best Head for a Rule

When using the subroutine REFINERULE, as given in Algorithm 9, for refining a rule, the subroutine FINDBESTHEAD is used to determine the multi-label head, which results in the best performance, regarding the refined rule's body. In theory, this requires to evaluate the multi-label heads, which result from all possible label combinations. However, as discussed in Chapter 3, such an exhaustive search through the label space suffers from an exponential computational complexity and therefore is not feasible, if many labels are available. For this reason, the proposed algorithm prunes searches according to the anti-monotonicity property, respectively the properties of decomposable evaluation functions, which are formally defined in Section 3.2 and Section 3.3. In order to perform as efficient as possible, the subroutine FINDBESTHEAD must be able to decide, whether an anti-monotonous or a decomposable evaluation metric is given. Algorithm 12 illustrates, how said subroutine opts for either executing the subroutine PRUNEDSEARCH or DECOMPOSITE, depending on the given evaluation function δ, as well as the used evaluation and averaging strategies. The operation of these subroutines is discussed in the following two subsections. Because the aim of the proposed algorithm is to learn multi-label head rules rather than single-label head rules, both subroutines PRUNEDSEARCH and DECOMPOSITE prefer heads, that consist of more label attributes, over heads with fewer attributes. In Chapter 5, selected multi-label evaluation metrics are examined in terms of whether they meet the properties of anti-monotonicity, respectively decomposability. Metrics, which neither fulfill the properties of anti-monotonicity, nor the properties of decomposability, are not supposed to be used by the proposed algorithm.

Require: Original training data set T, current training data set $T_{current}$,
current rule r, targets G, evaluation function δ,
whether to use rule-dependent or rule-independent evaluation strategy,
the averaging strategy to use

1 $r.head = \emptyset$
2 **if** δ is decomposable using the given evaluation and averaging strategy **then**
3 **return** DECOMPOSITE$(T, T_{current}, r, G, \delta)$
4 **else**
5 **return** PRUNEDSEARCH$(T, T_{current}, r, G, \delta, \emptyset)$

Algorithm 12: Algorithm FINDBESTHEAD for finding the best multi-label head for a given rule's body

4.2.1 Pruning based on Anti-Monotonicity

The subroutine PRUNEDSEARCH, which is shown in Algorithm 13, performs a depth-first search for finding the best multi-label head, regarding a given rule r. In theory, using a breadth-first search is feasible as well. The targets G specify the types of label attributes, which are allowed to be contained by a rule's head. If the target $G = \{1\}$ is used, only attributes of the form $\hat{y}_i = 1$, which predict the presence of labels, are taken into consideration. When using the targets $G = \{0, 1\}$ instead, label attributes of the form $\hat{y}_i = 0$, which predict the absence of labels, are considered as well. The search algorithm recursively adds additional label attributes to the initially empty head of the given rule and keeps track of the head, which reaches the highest performance. As it can be seen in Algorithm 13, the evaluation of unpromising label combinations is avoided in two ways: On the one hand, when adding an additional label attribute causes the performance of a rule to decrease, the subroutine PRUNEDSEARCH is not executed recursively any further. This is based on the anti-monotonicity property, which states, that by adding additional attributes, the best possible performance cannot be reached (cf. Section 3.2). On the other hand, heads, which have already been evaluated, are added to the (initially empty) set H. This allows to prevent equivalent heads from being unnecessarily evaluated again in later iterations. Note, that heads are considered to be equivalent, if they contain the same label attributes, regardless of their values and

order. By excluding supersets of the heads, which are already contained in H, from being evaluated in later iterations as well, it is ensured, that pruned heads are left out in later iterations (cf. Algorithm 13, line 10). If the targets $G = \{0, 1\}$ are used, Algorithm 13 determines for each potential label attribute, whether predicting the presence or absence of the corresponding label results in a better performance (cf. Algorithm 13, line 12). Moreover, labels, which are used by the conditions in a rule's body, are not available for being added to its head (cf. Algorithm 13, line 2).

Require: Original training data set T, current training data set $T_{current}$,
current rule r, targets G, evaluation function δ,
already considered heads H
whether to use rule-dependent or rule-independent evaluation strategy,
the averaging strategy to use

```
1  r_best = r
2  for each label λ_i not already contained in r.body do
3     r_current = r
4     r_current.h = −∞
5     for each target t ∈ G do                ▷ For each label, all targets are taken into consideration
6        ŷ_i = t                              ▷ Currently considered label attribute
7        if label attribute ŷ_i is not already in r.head then
8           r_refined = r
9           r_refined.head ∪ ŷ_i              ▷ Add label attribute to head
10          if no head in H is a subset of r_refined.head then   ▷ Prunes the evaluation of label combinations
11             r_refined.h = EVALUATERULE(T, T_current, r_refined, δ)
12             if r_refined.h > r_current.h then   ▷ Determines the best prediction for each label
13                r_current = r_refined
14    if r_current.h ≠ −∞ then                 ▷ If label combination has not been pruned
15       if r_current.h ≥ r_best.h then        ▷ Recursively add label attributes, unless performance decreased
16          r_rec = PRUNEDSEARCH(r_current, G, T, δ, H)
17          if r_rec.h > r_best.h or (r_rec.h == r_best.h and |r_rec.head| > |r_best.head|) then
18             r_best = r_rec                 ▷ Heads with more label attributes are prefered
19       H = H ∪ r_current.head
20 return rule r_best with best head
```

Algorithm 13: Algorithm PRUNEDSEARCH, which performs a pruned search through the label space, according to the properties of anti-monotonicity

The depth-first search, which is performed by Algorithm 13, can be visualized as a search tree, similar to the one shown in Figure 1. The nodes of such a tree correspond to the evaluation of particular label combinations, whereas the edges correspond to adding an additional label attribute to the head, which is represented by the preceding node. Because heads are prevented from being evaluated multiple times, the search tree is unbalanced.

4.2.2 Exploiting Decomposable Evaluation Metrics

If a decomposable evaluation metric is used for evaluating the performances of multi-label head rules, no deep search for finding the best multi-label head for a given rule's body is necessary. Instead, determining the best multi-label head comes at linear costs in such case, because it can be derived from the performances of all potential single-label head rules. Given n labels, only n, respectively $2 \cdot n$, single-label head rules must be evaluated, depending on whether the targets $G = \{1\}$ or $G = \{0, 1\}$ are used. The best multi-label head rule finally results from the single-label head rules, which reach the maximum performance. If only one single-label head rule reaches the best performance, according to the properties

of decomposable evaluation metrics, it is guaranteed, that no rule, which contains containing multiple label attributes in its head, can reach that performance as well (cf. Section 3.3). Instead, if multiple single-label head rules reach the best performance, the best multi-label head results from the label attributes, which are contained in those single-label head rules' heads. The subroutine DECOMPOSITE, which is shown in Algorithm 14 below, illustrates how these properties can be exploited in order to efficiently determine the best multi-label head, regarding the body of a given rule r. It measures the performance of all potential single-label head rules and keeps track of those rules, which reach the performance, which is known to be the current maximum. If a rule outperforms that performance, the resulting multi label head is replaced by the rule's single-label head (cf. Algorithm 14, line 12). If a rule reaches the performance, which is the maximum so far, the label attribute, which is contained by its head, is added to the resulting multi-label head (cf. Algorithm 14, line 14). The process continues until all potential single-label head rules, depending on the given targets G, have been considered. When using the targets $B = \{0, 1\}$, it is ensured that only one label attribute, which corresponds to the same label, is contained by a head (cf. Algorithm 14, line 10). Also, label attributes, which correspond to the labels, a rule's body depends on, are excluded from being added to its head (cf. Algorithm 14, line 2).

Require: Original training data set T, current training data set $T_{current}$,
current rule r, targets G, evaluation function δ,
whether to use rule-dependent or rule-independent evaluation strategy,
the averaging strategy to use

```
1  r_best = r
2  for each label λ_i not already contained in r.body do
3      r_current = r
4      r_current.h = −∞
5      for each target t ∈ G do                ▷ For each label, all targets are taken into consideration
6          ŷ_i = t                             ▷ Currently considered label attribute
7          if label attribute ŷ_i is not already in r.head then
8              r_single = ŷ_i ← r.body
9              r_single.h = EVALUATERULE(T, T_current, r_single, δ)
10             if r_single.h > r_current.h then    ▷ Determines the best prediction for each label
11                 r_current = r_single
12     if r_current.h > r_best.h then    ▷ If best performance is outperformed, the previous head is discarded
13         r_best = r_current
14     else if r_current.h = r_best.h then    ▷ If best performance is reached, label attribute is added to head
15         r_best.head ∪ r_current.head
16 return rule r_best with best head
```

Algorithm 14: Algorithm DECOMPOSITE, which exploits the properties decomposable evaluation metrics to determine the best possible multi-label head for a specific rule

4.3 Measuring the Performance of Multi-Label Head Rules

Algorithm 13 and 14 both depend on the subroutine EVALUATERULE for measuring the performances of multi-label head rules, in order to be able to compare them to each other. The evaluation of an individual rule depends on the following input parameters: The evaluation function δ, the averaging strategy, which should be used, and whether a rule-dependent or rule-independent evaluation strategy should be utilized (cf. Section 3.1). The subroutine EVALUATERULE, which is shown in Algorithm 15 below, measures the performance of a given multi-label head rule r on the training data set T, depending on these parameters. The evaluation strategy, which should be used, is taken into account by the subroutine GETRELEVANTLABELS. It returns the labels, which should be taken into account by the performance eval-

uation. Based on these labels, the performance is calculated by one of the subroutines MICROAVERAGING, LABELBASEDAVERAGING, EXAMPLEBASEDAVERAGING or MACROAVERAGING.

Require: Original training data set T, current training data set $T_{current}$,
rule r, evaluation function δ,
whether to use rule-dependent or rule-independent evaluation strategy,
the averaging strategy to use

```
1 L = getRelevantLabels(r)
2 if use micro-averaging then
3     microAveraging(T, T_current, r, δ, L)
4 else if use label-based averaging then
5     labelBasedAveraging(T, T_current, r, δ, L)
6 else if use example-based averaging then
7     exampleBasedAveraging(T, T_current, r, δ, L)
8 else if use macro-averaging then
9     macroAveraging(T, T_current, r, δ, L)
```

Algorithm 15: Algorithm EVALUATERULE for measuring the performance of a multi-label head rule

In order to be able to compare different multi-label head rules with each other, it is useful to take additional criteria besides solely relying on the heuristic values, which are calculated according to the given evaluation function, averaging strategy and evaluation strategy, into consideration. However, in favor of simplicity, this is not shown in Algorithm 15. Some criteria can be taken from heuristics, which turned out to be useful in separate-and-conquer rule learning for solving single- and multi-class classification problems. For example, as separate-and-conquer algorithms aim at iteratively cover the examples of a training data set, it is needless to induce rules, that do not cover any of these examples, although they may reach very high performances according to some evaluation functions. Furthermore, a strategy for comparing rules, which reach the same performance, is required. This is often referred to as "tie breaking". Common approaches for handling rules, which evaluate to equal performances, are based on preferring those, that cover more true positives or contain fewer conditions in their bodies. If the performances of multiple rules are equal, even when taking additional criteria into consideration, one of these rules must be chosen randomly [Janssen, 2012].

Require: Rule r, whether to use rule-dependent or rule-independent evaluation strategy

```
1 if use rule-dependent evaluation strategy then
2   return {λ_i | λ_i ∈ 𝕃 ∧ ŷ_i ∈ r.head}      ▷ Return labels, which are predicted by the given rule
3 else
4   return {λ_i | λ_i ∈ 𝕃}                      ▷ Return all available labels
```

Algorithm 16: Algorithm GETRELEVANTLABELS for retrieving all relevant labels according to the used evaluation strategy

Prior to measuring a multi-label head rule's performance, the subroutine GETRELEVANTLABELS, which is shown in Algorithm 16, is used to retrieve the labels, which must be taken into account by the evaluation. The labels, which are returned by said algorithm, depend on whether the rule-dependent or rule-independent evaluation strategy should be used. It is based on the definitions of both variants, as given in Section 3.1: If a rule-dependent evaluation strategy should be used, only the labels, which are predicted by the given rule r are returned. If a rule-independent evaluation strategy should be used instead, all labels of the respective data set T are returned, regardless of the given rule.

In Algorithm 17, the operation of the subroutine AGGREGATE is shown. It is used by each of the subroutines MICROAVERAGING, LABELBASEDAVERAGING, EXAMPLEBASEDAVERAGING and MACROAVERAGING in order to

build confusion matrices. The algorithm decides, whether the prediction of a rule r for a given training example and label is considered as a true positive, false positive, true negative or false negative. A confusion matrix C, which is passed to said algorithm as an argument, is altered accordingly. According to Algorithm 17, the output of the subroutine AGGREGATE depends on whether the given example is covered by the rule, or not. On the one hand, if the example is not covered, the rule's head is not applied, resulting in no labels being set. Therefore, in such case relevant labels are counted as false negatives and irrelevant labels contribute to the true negatives of a confusion matrix. On the other hand, if the example is covered by the rule, the outcome depends on the rule's prediction for the given label. If the label is predicted correctly, the true positives of the given confusion matrix are increased. If the prediction is incorrect, this affects the false positives of the confusion matrix instead.

Require: Rule r, training example (X_j, Y_j), label λ_i, confusion matrix C

```
1  if r covers X_j then
2      if h.head contains a label attribute ŷ_i then
3          if ŷ_i ≠ y_i ∈ Y_j then
4              C.fp+=1                                   ▷ Prediction is incorrect
5          else
6              C.tp+=1                                   ▷ Prediction is correct
7      else                  ▷ Does only occur when using label-independent evaluation strategy
8          if y_i ∈ Y_j is set then
9              C.fp+=1                                   ▷ Relevant label predicted as irrelevant
10         else
11             C.tp+=1                                   ▷ Irrelevant label predicted as irrelevant
12 else
13     if y_i ∈ Y_j is set then
14         C.fn+=1
15     else
16         C.tn+=1
```

Algorithm 17: Algorithm AGGREGATE for aggregating the true positives, false positives, true negatives and false negatives of a confusion matrix

The way Algorithm 17 treads true positives, false positives, true negatives and false negatives, contradicts the definition, which is given in Equation 2.5. That definition is exclusively meant to be used for evaluating the performance of a learned model on a test data set. When evaluating the performances of individual rules during the rule induction process by using Algorithm 17, correctly predicted labels are always counted as true positives, regardless of whether the label is relevant or irrelevant, whereas incorrectly predicted labels are always counted as false positives. The reason for using this divergent semantic is, that all label attributes of a rule's head should be handled equally, regardless of whether they predict the presence or absence of a label. If the traditional semantic according to Equation 2.5 would be used instead, the evaluation of individual label attributes would depend on whether they are of the form $\hat{y}_i = 1$ or $\hat{y}_i = 0$. For example, when using the precision metric, which is exclusively calculated from true positives and false positives, label attributes of the form $\hat{y}_i = 0$ would never have a positive impact on the overall performance of a rule, as they only produce true negatives and false negatives.

4.3.1 Micro-Averaging

When using micro-averaging, the performance of a multi-label head rule is calculated using the soubroutine MICROAVERAGING as shown in Algorithm 18. The operation of the algorithm corresponds to the definition of micro-averaging given in Equation 2.8.

Require: Original training data set T, current training data set $T_{current}$, rule r, evaluation function δ, relevant labels L

1 C = (0,0,0,0) ▷ Initialize global confusion matrix
2 **for each** example $(X_j, Y_j) \in T$ **do**
3 **if** r does not cover X_j or not all labels $\hat{y}_i \in r.head$ are set in $Y_j \in T_{current}$ **then**
4 **for each** relevant label $\lambda_i \in L$ **do**
5 AGGREGATE$(r, (X_j, Y_j), \lambda_i, C)$
6 $h = \delta(C)$ ▷ Apply evaluation function on confusion matrix
7 **return** performance h

Algorithm 18: Algorithm MICROAVERAGING for measuring the performance of a multi-label head rule using micro-averaging

As it can be seen in Algorithm 18, the current training data set $T_{current}$ is used to check, whether labels are already predicted by previously induced rules, or not (cf. Algorithm 18, line 3). Covered examples, for which all labels, that are predicted by a rule, are marked as already set (cf. Algorithm 5, line 3), are not taken into consideration for building the aggregated confusion matrix. In the subroutines LABELBASEDAVERAGING, EXAMPLEBASEDAVERAGING and MACROAVERAGING, which are discussed in the following, similar checks can be found. All of them correspond to a mechanism, which is used in the original separate-and-conquer algorithm by Loza Mencía and Janssen [2015] in order to prevent identical rules from being learned in subsequent iterations of the algorithm. In theory, this might occur when no examples are removed from the training data set after inducing a new rule, because the examples' label vectors are not considered to be covered to a sufficient extend by subroutine GETREADDSET. As the original algorithm focuses on learning single-label head rules, only one label is considered at once. This enables it to temporarily remove the examples, for which the current label is marked as already predicted, from the training data set (cf. Algorithm 4, line 3). However, this strategy cannot be used by the algorithm, which is proposed in this work, because the induction of multi-label head rules requires to consider all available labels simultaneously. As an alternative, instances, for which a rule does not predict yet unset labels, are not taken into account for measuring the rule's performance.

4.3.2 Label-based Averaging

Algorithm 19 illustrates the operation of the subroutine LABELBASEDAVERAGING, which is used for measuring the performance of multi-label head rules, when using label-based averaging. This corresponds to the averaging strategy, which is defined in Equation 2.9.

Require: Original training data set T, current training data set $T_{current}$, rule r, evaluation function δ, relevant labels L

1 $h = 0$
2 **for each** relevant label $\lambda_i \in L$ **do**
3 $C = (0,0,0,0)$ ▷ Initialize confusion matrix for each label
4 **for each** example $(X_j, Y_j) \in T$ **do**
5 **if** r does not cover X_j or not all labels $\hat{y}_i \in r.head$ are set in $Y_j \in T_{current}$ **then**
6 AGGREGATE$(r, (X_j, Y_j), \lambda_i, C)$
7 $h += \delta(C)$ ▷ Apply evaluation function on confusion matrix
8 $h = h/|L|$ ▷ Average performance label-wise
9 **return** performance h

Algorithm 19: Algorithm LABELBASEDAVERAGING for measuring the performance of a multi-label head rule using label-based averaging

The algorithm, which is shown above, first calculates an individual heuristic value for each relevant label. These values result from building one confusion matrix per label using the subroutine AGGREGATE and applying the given evaluation function δ on it. The true positives, false positives, true negatives, and false negatives of each confusion matrix are counted by applying the rule r on all training examples and aggregating the predictive result for the corresponding label. Finally, the given rule's overall performance calculates as the arithmetic mean of the heuristic values, which have been obtained for each relevant label. Such as discussed in the previous section, examples, for which no yet unset labels are predicted, are excluded from the performance evaluation.

4.3.3 Example-based Averaging

When using example-based averaging, the subroutine EXAMPLEBASEDAVERAGING is used for measuring the performance of a rule. It is similar to the subroutine LABELBASEDAVERAGING, which is discussed in the previous section, but instead of aggregating true positives, false positives, true negatives and false negatives label-wise, an example-wise aggregation is used. This corresponds to the definition of said averaging strategy as given in Equation 2.9. Algorithm 20 illustrates how the example-based performance of a multi-label rule r is measured. It calculates a heuristic value for each training example by applying the evaluation function δ on confusion matrices, which are built from the given rule's prediction for all relevant labels. By averaging the heuristic values, which have been obtained for each example, the overall performance of the given multi-label head rule is finally calculated. Similar to Algorithm 18 and Algorithm 19, if all labels, which are predicted by a rule, are marked as already predicted by previously induced rules, the respective example is excluded from the performance evaluation.

Require: Original training data set T, current training data set $T_{current}$,
rule r, evaluation function δ, relevant labels L

1 $h = 0$
2 **for each** example $(X_j, Y_j) \in T$ **do**
3 **if** r does not cover X_j or not all labels $\hat{y}_i \in r.head$ are set in $Y_j \in T_{current}$ **then**
4 $C = (0, 0, 0, 0)$ ▷ Initialize confusion matrix for each example
5 **for each** relevant label $\lambda_i \in L$ **do**
6 AGGREGATE$(r, (X_j, Y_j), \lambda_i, C)$
7 $h \mathrel{+}= \delta(C)$ ▷ Apply evaluation function on confusion matrix
8 $h = h/|T|$ ▷ Average performance example-wise
9 **return** performance h

Algorithm 20: Algorithm EXAMPLEBASEDAVERAGING for measuring the performance of a multi-label head rule using example-based averaging

4.3.4 Macro-Averaging

The subroutine MACROAVERAGING, which is shown in Algorithm 21 below, is used for measuring the performance of a multi-label head rule r using macro-averaging as defined in Equation 2.11. As it can be seen in the algorithm, for each training example and relevant label an individual confusion matrix is created. It specifies, whether the prediction of the rule r for a particular example and label combination is considered as a true positive, false positive, true negative or false negative. By applying the evaluation function δ on each of these confusion matrices, a heuristic value is obtained for each example and relevant label. The overall performance of the rule finally calculates as the label- and example-wise arithmetic mean of all of these values. According to Equation 2.11, averaging the obtained values is commutative, i.e. first calculating the label-wise arithmetic mean and then averaging the values example-wise is computationally equal to performing the calculation the other way round.

Require: Original training data set T, current training data set $T_{current}$, rule r, evaluation function δ, relevant labels L

```
1  h = 0
2  for each example (X_j, Y_j) ∈ T do
3      if r does not cover X_j or not all labels ŷ_i ∈ r.head are set in Y_j ∈ T_current then
4          h_j = 0
5          for each relevant label λ_i ∈ L do
6              C = (0,0,0,0)                    ▷ Initialize confusion matrix for each example and label
7                  AGGREGATE(r, (X_j, Y_j), λ_i, C)
8              h_j += δ(C)                      ▷ Apply evaluation function on confusion matrix
9          h += h_j/|L|                         ▷ Average performance label-wise
10 h = h/|T|                                    ▷ Average performance example-wise
11 return performance h
```

Algorithm 21: Algorithm MACROAVERAGING for measuring the performance of a multi-label head rule using macro-averaging

4.4 Application of Multi-Label Head Rules

The prediction process for applying a model, which has been deduced from a training data set by using the proposed algorithm, on an unknown test example is similar to the prediction process of the original algorithm by Loza Mencía and Janssen [2015] as discussed in Section 2.2.3. In both cases, the rules, which are contained by a learned multi-label decision list, are applied to the test example in the order of induction until all rules are processed or a stopping rule is encountered. However, as the algorithm, which is proposed is the present work, allows to induce multi-label head rules, several labels can be set by a single rule. In line 4 of Algorithm 22, the loop, which is used to set all labels, which correspond to the label attributes of a multi-label head, is shown. Similar to the original prediction algorithm given in Algorithm 7, labels are only set, if they have not already been set by previously processed rules. This prevents the predictions of rules from being altered by other rules, which have been learned at a later point of the rule induction process. Algorithm 22 continues until all labels of the test example are set or until no rules remain in the decision list. Labels, which remain unset after the prediction process is finished, are assumed to be irrelevant.

Require: Test example X, multi-label decision list R

```
1  Ŷ = ⟨?, ..., ?⟩
2  for each rule r in decision list R do
3      if r covers X then
4          for each label attribute ŷ_i ∈ r.head do
5              apply label attribute ŷ_i on Ŷ if corresponding value in Ŷ is unset
6          if r is marked as stopping rule or Ŷ is complete then
7              assume all remaining labels in Ŷ to be irrelevant
8              return prediction Ŷ
9  assume all remaining labels in Ŷ to be irrelevant
10 return prediction Ŷ
```

Algorithm 22: Algorithm for predicting the label vector of a test example, based on the multi-label head rules of a multi-label decision list

5 Anti-Monotonicity and Decomposability of Multi-Label Evaluation Metrics

In this chapter, the evaluation functions, which are given in Section 2.3.3, are examined in terms of anti-monotonicity and decomposability. In Section 5.1, the evaluation functions are examined with respect to the rule-dependent evaluation strategy, which is introduced in Section 3.1. In Section 5.2 the rule-independent evaluation strategy is considered accordingly. In both sections, for each evaluation function and averaging strategy, it is mathematically proved or disproved, whether the definition of anti-monotonicity, respectively decomposability, is met, or not. For reasons of simplicity –if not stated otherwise –, the monotonicity property according to Definition 3.2 is assumed to not be fulfilled by evaluation functions. In each case, that definition can easily be disproved by giving a simple counterexample. Throughout the proofs, which are given in the present chapter, evaluation functions are considered as surjections, which map the predictive results of a rule r on a data set T to a heuristic value $h \in [0, 1]$:

$$\delta : r, T \rightarrow \mathbb{R} \tag{5.1}$$

According to Section 2.3.3, the evaluation functions are defined as functions, which operate on the elements of confusion matrices. In order to denote these elements, the following surjections $r, T \rightarrow \mathbb{N}$ are used throughout this chapter to retrieve the number of true positives, false positives, true negatives and false negatives a rule r covers on a data set T. In accordance with Chapter 5, these elements are calculated as shown in Algorithm 17, instead of using the definition in Equation 2.5. The indices i and j specify the label λ_i and example X_j, the value should be obtained for.

$$\begin{aligned} TP_i^j : r, T \rightarrow \mathbb{N} \qquad FP_i^j : r, T \rightarrow \mathbb{N} \\ TN_i^j : r, T \rightarrow \mathbb{N} \qquad FN_i^j : r, T \rightarrow \mathbb{N} \end{aligned} \tag{5.2}$$

According to the shorthand notations, which are introduced in Equation 2.6, the following surjections $r, T \rightarrow \mathbb{N}$ are used to denote the number of relevant and irrelevant labels according to the ground truth, respectively to a particular prediction:

$$\begin{aligned} P_i^j(r, T) &:= TP_i^j(r, T) + FN_i^j(r, T) \qquad N_i^j(r, T) := FP_i^j(r, T) + TN_i^j(r, T) \\ p_i^j(r, T) &:= TP_i^j(r, T) + FP_i^j(r, T) \qquad n_i^j(r, T) := TN_i^j(r, T) + FN_i^j(r, T) \end{aligned} \tag{5.3}$$

In order to simplify the notation of the mathematical proofs, the following function is used throughout this chapter. It returns the subset of a data set T, that only contains the instances (X_j, Y_j), which are covered by a given rule r:

$$C(r, T) := \left\{ (X_j, Y_j) \in T \mid r \text{ covers } X_j \right\} \tag{5.4}$$

In many cases, the readability of mathematical proof is further increased by omitting the parameters r and T of the functions, which are given above, in favor of simplicity. This is possible, if only one rule r and one data set T take part in an equation. However, if multiple rules are used in a single equation, the subscript notation, which is shown below, is used to unambiguously denote the rule $\hat{Y} \leftarrow B$ and data set T, which should be used for evaluating the left-hand expression f.

$$f|_{\hat{Y} \leftarrow B, T} \tag{5.5}$$

5.1 Rule-dependent Evaluation

In the following subsections, the evaluation functions precision, recall, hamming accuracy, F-measure and subset accuracy are examined in terms of anti-monotonicity and decomposability regarding the rule-dependent evaluation strategy. When using said evaluation strategy, according to Section 3.1, only the labels, which are predicted by a rule, are taken into account by a performance evaluation. In some of the following subsections, the following lemma is examined in order to prove the definition of decomposability to be met. If said lemma holds for a particular evaluation function and averaging strategy, the definition of decomposability is implied to be met as well.

Lemma 5.1: *Let $\hat{Y} \leftarrow B$ denote a multi-label head rule, consisting of a body B and a head $\hat{Y}$. Given a particular averaging strategy and using a rule-dependent evaluation strategy, an evaluation function δ is decomposable according to Definition 3.3, if the performance of the rule is equal to the arithmetic mean of the performances, which are obtained by considering each label attribute $\hat{y}_i \in \hat{Y}$ individually:*

$$\delta(\hat{Y} \leftarrow B) = \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \leftarrow B} \delta(\hat{y}_i \leftarrow B)$$

Proof: Such as all Pythagorean means, the arithmetic mean has two well-known properties, which are referred to as “averaging” and “value preservation” in the following [Heath, 1921]. Because of these properties, the definition of decomposability is met, if the performance of a rule calculates as shown in Lemma 5.1 above. On the one hand, the precondition, which is given in Definition 3.3 i), is met because of the arithmetic mean’s averaging property:

$$min(x_1, ..., x_n) < \frac{1}{n} \cdot \sum_{i=1}^{n} x_i < max(x_1, ..., x_n) \tag{5.6}$$
Averaging

On the other hand, the precondition, which is given in Definition 3.3 ii), is met because of the value preservation property:

$$\frac{1}{n} \cdot \sum_{i=1}^{n} x_i = x_1 = ... = x_n \text{ , with } x_1 = ... = x_n \tag{5.7}$$
Value preservation

■

5.1.1 Precision

According to the definition, which is given in Equation 2.12, the precision metric measures the percentage of correctly predicted labels (true positives) among all labels, which are predicted by a rule (true positives and false positives). Due to the fact, that the precision metric only takes true positives and false positives into account, the performance of a rule solely depends on the examples it covers. This is, because for covered examples the predicted labels are counted as true positives or false positives, whereas the labels of uncovered examples contribute to the true negatives and false negatives (cf. Algorithm 17). In the following subsections, it is proved, that the precision metrics fulfills the definition of decomposability according to Definition 3.3, if the rule-dependent evaluation strategy is utilized. As it is shown in the individual subsections, the properties of decomposability are met, regardless of whether micro-averaging, label-based averaging or macro-averaging is used.

5.1.1.1 Micro-Averaging

In the following, it is proved, that micro-averaged precision meets the requirements of decomposability, when using the rule-dependent evaluation strategy. In Figure 3 of Chapter 3, an example, which uses micro-averaged precision for measuring the rule-dependent performances of multi-label head rules, is given. The proof is based on how the micro-averaged precision of a single-label head rule is calculated. Equation 5.8 shows such calculation for a rule, which contains a single label attribute $\hat{y}_i$ in its head. As the equation reveals, the precision of a single-label head rule calculates as the fraction of true positives among the number of all predicted labels. As the prediction of a rule is only applied to covered examples, the number of predictions equals the number of examples, which are covered by the rule's body B. According to the shorthand notation, which is introduced in Equation 5.4, the number of covered examples is denoted as $|C|$.

$$\begin{aligned} \delta_{prec,mm,\not\perp}(\hat{y}_i \leftarrow B, T) &= \frac{\sum_j TP_i^j}{\sum_j p_i^j} \text{ , with } \sum_j p_i^j = |C| \\ &= \frac{\sum_j TP_i^j}{|C|} \end{aligned} \tag{5.8}$$

Proof of Decomposability: The proof, which is given in Equation 5.9 below, shows Lemma 5.1 to be hold, when using micro-averaged precision together with the rule-dependent evaluation strategy for measuring the performance of a multi-label head rule $\hat{Y} \leftarrow B$. As required by Lemma 5.1, the performance calculation can be rewritten in terms of measuring the performances of single-label head rules $\hat{y}_i \leftarrow B$ with $\hat{y}_i \in \hat{Y}$ and averaging the obtained results afterwards by using the arithmetic mean operation. Rewriting the equation is possible, because all of the single-label head rules share the same body B and therefore cover the same number of examples $|C|$. This observation corresponds to Equation 5.8, which is given above.

$$\begin{aligned} \delta_{prec,mm,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j TP_i^j}{\sum_{\hat{y}_i \in \hat{Y}} \sum_j p_i^j} \text{ , with } \sum_j p_i^j = |C| \text{ , } \forall i \\ &= \frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j TP_i^j}{|\hat{Y}| \cdot |C|} \\ &= \frac{1}{|\hat{Y}|} \cdot \frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j TP_i^j}{|C|} \\ &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \frac{\sum_j TP_i^j}{|C|} \qquad \triangleright \text{ c.f (5.8), last line} \\ &\equiv \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \delta_{prec,mm,\not\perp}(\hat{y}_i \leftarrow B, T) \end{aligned} \tag{5.9}$$

Equation 5.9 proves Lemma 5.1 to be met, when micro-averaged precision is used together with the rule-dependent evaluation strategy. As a result, said evaluation function is implied to be decomposable according to Definition 3.3. Furthermore – because of Lemma 3.1 –, the evaluation function can be considered to be anti-monotonous according to Definition 3.1 as well. ■

5.1.1.2 Label-based Averaging

When using label-based averaging together with the rule-dependent evaluation strategy, the calculation of a multi-label head rule's performance is in accordance with Lemma 5.1 per se. This is, because of said averaging strategy's definition, which is given in Equation 2.10. It states, that a label-based performance calculates as the label-wise arithmetic mean of the performances, which are obtained for each label individually. Such as Lemma 5.1 requires, this corresponds to single-label head rules $\hat{y}_i \leftarrow B$ with $\hat{y}_i \in \hat{Y}$. However, the proof, which is given in the following, shows, that utilizing label-based averaging for measuring the precision of a multi-label head rule $\hat{Y} \leftarrow B$ is even equivalent to using micro-averaging, as discussed in the previous section.

Proof of Decomposability: Equation 5.12 is based on the fact, that the precision of a single-label head rule $\hat{y}_i \leftarrow B$ is calculated according to Equation 5.8, regardless of whether micro-averaging or label-based averaging is used. This enables to rewrite the calculation of label-based precision in terms of using micro-averaged precision, as shown in the second line of Equation 5.10. As the rewritten term equals the last line of Equation 5.9, label-based precision and micro-averaged precision can be considered to be equivalent when using the rule-dependent evaluation strategy.

$$\begin{aligned}
\delta_{prec,mM,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \frac{\sum_j TP_i^j}{\sum_j P_i^j} \text{ , with } \sum_j p_i^j = |C| \text{ , } \forall i \qquad \triangleright \text{ cf. (5.8), first line} \\
&\equiv \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \delta_{prec,mm,\not\perp}(\hat{y}_i \leftarrow B, T) \qquad \triangleright \text{ cf. (5.9), last line} \\
&\equiv \delta_{prec,mm,\not\perp}(\hat{Y} \leftarrow B, T)
\end{aligned} \tag{5.10}$$

Equation 5.10 shows, that label-based precision is equivalent to micro-averaged precision, when using the rule-dependent evaluation strategy. As the latter variant is proved to be decomposable in Section 5.1.1.1, its label-based counterpart is implied to be decomposable as well. ■

5.1.1.3 Example-based Averaging

When using example-based averaging for measuring the performance of a rule, according to Equation 2.9, a heuristic value is calculated for each of the data set's examples at first. Finally, the overall performance results from the arithmetic mean of all obtained values. Equation 5.11 shows, how the example-based precision of a single-label head rule $\hat{y}_i \leftarrow B$ is calculated according to that definition. As only one label takes part in the equation, a single prediction is made for each covered example. For examples, which are not covered by the rule, no predictions are made at all. Consequently, the heuristic values,

which are obtained for uncovered examples, always evaluate to 0. Because of this, the calculation can be rewritten to solely depend on the number of true positives, as it is shown in Equation 5.11 below.

$$\begin{aligned}\delta_{prec,Mm,\not\perp\!\!\!\perp}(\hat{y}_i \leftarrow B, T) &= \frac{1}{m} \cdot \sum_j \frac{TP_i^j}{p_i^j} \text{ , with } \begin{smallmatrix} p_i^j=1 \text{ , } \forall j(X_j \in C) \\ TP_i^j=0 \text{ , } \forall j(X_j \notin C) \end{smallmatrix} \\ &= \frac{1}{m} \cdot \sum_j TP_i^j \\ &= \sum_j \frac{TP_i^j}{m}\end{aligned} \tag{5.11}$$

Proof of Decomposability: Equation 5.12 shows, how the example-based precision of a multi-label head rule $\hat{Y} \leftarrow B$, which contains an arbitrary number of label attributes $\hat{y}_i$ in its head $\hat{Y}$, is calculated. By rewriting the equation to solely depend on true positives – similar as shown in Equation 5.11 above –, it can be proved that the performance calculation is in accordance with Lemma 5.1. Rewriting the equation is based the following observation: On the one hand, for examples, which are covered by the body B, exactly $|\hat{Y}|$ predictions are made. This corresponds to number of label attributes, which are contained by the rule's head. On the other hand, if an example is not covered by the rule, no prediction is made and the corresponding heuristic value evaluates to 0.

$$\begin{aligned}\delta_{prec,Mm,\not\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{m} \cdot \sum_j \frac{\sum_{\hat{y}_i \in \hat{Y}} TP_i^j}{\sum_{\hat{y}_i \in \hat{Y}} p_i^j} \text{ , with } \begin{smallmatrix} \sum_{\hat{y}_i \in \hat{Y}} p_i^j = |\hat{Y}| \text{ , } \forall j(X_j \in C) \\ \sum_{\hat{y}_i \in \hat{Y}} TP_i^j = 0 \text{ , } \forall j(X_j \notin C) \end{smallmatrix} \\ &= \frac{1}{m} \cdot \sum_j \frac{\sum_{\hat{y}_i \in \hat{Y}} TP_i^j}{|\hat{Y}|} \\ &= \frac{1}{m} \cdot \sum_{\hat{y}_i \in \hat{Y}} \sum_j \frac{TP_i^j}{|\hat{Y}|} \\ &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \sum_j \frac{TP_i^j}{m} \qquad \triangleright \text{ c.f (5.11), last line} \\ &\equiv \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \delta_{prec,Mm,\not\perp\!\!\!\perp}(\hat{y}_i \leftarrow B, T)\end{aligned} \tag{5.12}$$

According to Equation 5.12, example-based precision can be rewritten in terms of measuring the performances of single-label head rules $\hat{y}_i \leftarrow B$ with $\hat{y}_i \in \hat{Y}$ and averaging the obtained heuristic values afterwards. This corresponds to the equation, which is given in Lemma 5.1. Because the lemma is shown to be met, it follows, that example-based precision is decomposable according to Definition 3.3. Moreover – because of Lemma 3.1 –, said evaluation function can also be considered to be anti-monotonous according to Definition 3.1. ■

5.1.1.4 Macro-Averaging

According to the definition of macro-averaging, which is given in Equation 2.11, when using said averaging strategy, a heuristic value is obtained for each example and label at first. All of the values, which are obtained in that way, are finally aggregated to a single performance by either first calculating the example-wise arithmetic mean and then calculating the label-wise arithmetic mean, or vice versa. The first of both orders is utilized in the following in order to show, that the use of macro-averaging is equivalent to the use of example-based averaging, as considered in Section 5.1.1.3. Such as all considerations, which are made in the current section, this corresponds to measuring the performance of multi-label head rules using the precision metric and utilizing the rule dependent evaluation strategy.

Proof of Decomposability: In Equation 5.13, the calculation of a multi-label head rule's performance, according to the definition of the precision metric and macro-averaging, is given. As already mentioned, the calculation is based on averaging first example-wise and then label-wise. Because for each example and label the performance either evaluates to 0 or 1, depending on whether a true positive is covered, or not, it is possible to rewrite the calculation to solely depend on true positives. As the second line of Equation 5.13 illustrates, the rewritten variant of the calculation is in accordance with Equation 5.12. Because of this, it follows, that macro-averaged precision is equivalent to example-based precision, when using the rule-dependent evaluation strategy.

$$
\begin{aligned}
\delta_{prec,MM,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \left(\frac{1}{m} \cdot \sum_j \frac{TP_i^j}{p_i^j} \right) \text{, with } \frac{TP_i^j}{p_i^j} = TP_i^j \text{ , } \forall i \forall j \\
&= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \sum_j \frac{TP_i^j}{m} \qquad \triangleright \text{ c.f (5.12), line 4} \\
&\equiv \delta_{prec,Mm,\not\perp}(\hat{Y} \leftarrow B, T)
\end{aligned}
\tag{5.13}
$$

Equation 5.13 shows, that macro-averaged precision is equivalent to example-based precision, when using the rule-dependent evaluation strategy. The example-based variant of the precision metric is proved to be decomposable in Section 5.1.1.3. Because of this, the its macro-averaged counterpart is implied to fulfill the properties of decomposability, which are given in Definition 3.3, as well. ■

5.1.2 Recall

The recall metric measures the percentage of labels, which are predicted to be relevant by a rule, among all labels, which are relevant according to the ground truth. According to Equation 2.13, this corresponds to the percentage of true positives among true positives and false negatives. In contrast to precision, when using the recall metric for measuring the performance of a rule, uncovered examples have an impact on the resulting performance. This is, because false negatives are taken into account by the performance evaluation. According to Algorithm 17, false negatives result from the relevant labels of the examples, which are not covered by a rule. As no predictions are made for these uncovered examples, their relevant labels are not predicted despite their relevance. In the following subsections, the recall metric is examined in terms of anti-monotonicity, respectively decomposability, when using micro-averaging, label-based averaging or macro-averaging together with the rule-dependent evaluation strategy.

5.1.2.1 Micro-Averaging

The proof, which is shown in the following, is based on the fact, that the micro-averaged recall of a multi-label head rule $\hat{Y} \leftarrow B$ can be calculated as the *mediant* of the performances, which are obtained for corresponding single-label head rules $\hat{y}_i \leftarrow B$ with $\hat{y}_i \in Y$. In mathematics, the mediant of two or more fractions $\frac{a_1}{b_1}, \ldots, \frac{a_n}{b_n}$ with $a_i \geq 0$ and $b_i > 0$ is defined as the fraction, which results from summing up the numerators and denominators of the given fractions [Bensimhoun, 2013]:

$$m\left(\frac{a_1}{b_1}, \ldots, \frac{a_n}{b_n}\right) = \frac{a_1 + \ldots + a_n}{b_1 + \ldots + b_n}, \text{ with } a_i, b_i \in \mathbb{N} \tag{5.14}$$
Mediant

A well-known property of the mediant, which is called “mediant inequality”, states, that the mediant strictly lies between the fractions it is calculated from. According to this property, the following inequality holds [Bensimhoun, 2013]:

$$min\left(\frac{a_1}{b_1}, \ldots, \frac{a_n}{b_n}\right) < m\left(\frac{a_1}{b_1}, \ldots, \frac{a_n}{b_n}\right) < max\left(\frac{a_1}{b_1}, \ldots, \frac{a_n}{b_n}\right) \tag{5.15}$$
Mediant inequality

As a special case, if all of the fractions $\frac{a_1}{b_1}, \ldots, \frac{a_n}{b_n}$ are equal, their mediant is equal to all of the fractions as well. With respect to the property of the arithmetic mean, which is introduced in Equation 5.7, this is referred to as the mediant’s “value preservation” property in the following.

$$m\left(\frac{a_1}{b_1}, \ldots, \frac{a_n}{b_n}\right) = \frac{a_1}{b_1} = \ldots = \frac{a_n}{b_n}, \text{ with } \frac{a_1}{b_1} = \ldots = \frac{a_n}{b_n} \tag{5.16}$$
Value preservation

In the following it is proved, that the properties, which are given in Equation 5.15 and 5.16, apply to the calculation of the micro-averaged recall, when using the rule-dependent evaluation strategy. As the mediant inequality corresponds to the property, which is given in Definition 3.3 i), whereas the mediant’s value preservation property corresponds to Definition 3.3 ii), this is sufficient to proof the definition of decomposability to be met.

Proof of Decomposability: Equation 5.17 shows, how the micro-averaging recall of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated, when using the rule-dependent evaluation strategy. In order to prove the mediant inequality according to Equation 5.15 to be met, it is assumed without loss of generality, that the performance of the single-label head rule $\hat{y}_b \leftarrow B$ with $\hat{y}_b \in \hat{Y}$ is the best among all single-label head rules, whose label attribute is contained in the multi-label head $\hat{Y}$. As the $\geqq$ operator indicates, in order to prove the value preservation property according to Equation 5.16 to be hold, the premise of the proof at hand can simply be adapted by assuming, that all single-label head rules reach the same performance.

$$\delta_{rec,mm,\not\perp}(\hat{Y} \leftarrow B, T) = \frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j TP_i^j}{\sum_{\hat{y}_i \in \hat{Y}} \sum_j P_i^j}, \text{ with } \frac{\sum_j TP_b^j}{\sum_j P_b^j} \geqq \frac{\sum_j TP_a^j}{\sum_j P_a^j}, \forall a \left(\hat{y}_a, \hat{y}_b \in \hat{Y} \wedge a \neq b\right) \tag{5.17}$$

As it can be seen in Equation 5.17, the micro-averaged recall of the multi-label head rule $\hat{Y} \leftarrow B$ is calculated by summing up the denominators and numerators of the fractions, which denote the performances of the corresponding single-label head rules. This corresponds to the definition of the mediant operation given in Equation 5.14. According to the premise of the proof, the single-head rule $\hat{y}_b \leftarrow B$ is considered to reach the best performance, whereas the remaining single-label head rules $\hat{y}_a \leftarrow B$ reach a lower performance.

In order to prove the properties of decomposability to be met, it must be shown, that the recall of the multi-label head rule $\hat{Y} \leftarrow B$ is less than the recall of the best single-label head rule $\hat{y}_b \leftarrow B$. This corresponds to Definition 3.3 i). In order to prove Definition 3.3 ii) to be met, it must be shown, that the performance of the multi-label head rule $\hat{Y} \leftarrow B$ is equal to the recall of all single-label head rules $\hat{y}_b \leftarrow B$ and $\hat{y}_a \leftarrow B$ accordingly.

$$\delta_{rec,mm,\not\perp}(\hat{Y} \leftarrow B, T) \leqq \delta_{rec,mm,\not\perp}(\hat{y}_b \leftarrow B, T)$$

$$\frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j TP_i^j}{\sum_{\hat{y}_i \in \hat{Y}} \sum_j P_i^j} \leqq \frac{\sum_j TP_b^j}{\sum_j P_b^j}$$

$$\frac{\sum_j TP_b^j}{\sum_j P_b^j} - \frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j TP_i^j}{\sum_{\hat{y}_i \in \hat{Y}} \sum_j P_i^j} \geqq 0 \tag{5.18}$$

$$\frac{\sum_{\hat{y}_i \in \hat{Y} \setminus \hat{y}_b} \left(\sum_j TP_b^j \cdot \sum_j P_i^j \right) - \sum_{\hat{y}_i \in \hat{Y} \setminus \hat{y}_b} \left(\sum_j TP_i^j \cdot \sum_j P_b^j \right)}{\sum_j P_b^j \cdot \sum_{\hat{y}_i \in \hat{Y}} \sum_j P_i^j} \geqq 0$$

In order to prove Equation 5.18 to be hold, it is rewritten in terms of a single fraction by converting both fractions, which take part in the original form of the equation, to like quantities. For this reason, the denominators of both fractions are multiplied with each other and the numerators are adapted accordingly. As a result, the equation can be proved to be hold by showing, that the fraction's numerator is positive – respectively equal to 0. The formula, which finally proves the assumption given in Equation 5.18 to be true, is shown in Equation 5.19 below. It is based on the proof's premise, which is originally introduced in Equation 5.17. By using cross-multiplication, the equation can be converted into an inequality – respectively an equality when proving 3.3 ii) to be met – similar to the numerator of the fraction, which is shown in Equation 5.18 above.

$$\frac{\sum_j TP_b^j}{\sum_j P_b^j} \geqq \frac{\sum_j TP_a^j}{\sum_j P_a^j}, \forall a \left(\hat{y}_a, \hat{y}_b \in \hat{Y} \wedge a \neq b \right) \qquad \triangleright \text{ c.f (5.17)}$$

$$\sum_j TP_b^j \cdot \sum_j P_a^j \geqq \sum_j TP_a^j \cdot \sum_j P_b^j, \forall a \left(\hat{y}_a, \hat{y}_b \in \hat{Y} \wedge a \neq b \right)$$

$$\sum_{\hat{y}_i \in \hat{Y} \setminus \hat{y}_b} \left(\sum_j TP_b^j \cdot \sum_j P_i^j \right) \geqq \sum_{\hat{y}_i \in \hat{Y} \setminus \hat{y}_b} \left(\sum_j TP_i^j \cdot \sum_j P_b^j \right) \qquad \triangleright \text{ Proofs last line of (5.18) to be hold}$$

(5.19)

As Equation 5.18 is shown to be hold, the recall metric is proved to meet the properties of decomposability according to Definition 3.3, when used together with micro averaging and the rule-dependent evaluation strategy. Because of Lemma 3.1, it can also be considered to be anti-monotonous in such case. ■

5.1.2.2 Label-based Averaging

As mentioned earlier, when using label-based averaging together with the rule-dependent evaluation strategy for measuring the performance of multi-label head rules, the utilized evaluation function can be considered to be decomposable per se. This is due to the definition of label-based averaging as given in Equation 2.10. The proof, which is shown in the following, confirms that.

Proof of Anti-Monotonicity: When using label-based averaging, the recall of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated by obtaining a heuristic value per label and averaging the results afterwards. If a rule-dependent evaluation is used, this corresponds to the calculation, which is shown in Equation 5.20.

$$\begin{aligned} \delta_{rec,mM,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \frac{\sum_j TP_i^j}{\sum_j P_i^j} \\ &\equiv \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \delta_{rec,mM,\not\perp}(\hat{y}_i \leftarrow B, T) \end{aligned} \tag{5.20}$$

As the calculation, which is shown above, corresponds to the equation, which is given as part of Lemma 5.1, label-based recall is implied to be decomposable when using the rule-dependent evaluation strategy. Note, that unlike label-based and micro-averaged precision, which are proved to be equivalent in Section 5.1.1, when using said evaluation strategy, label-based and micro-averaged recall are not equivalent. ■

5.1.2.3 Example-based Averaging

In the following it is shown, that example-based recall is neither decomposable, nor anti-monotonous, when using the rule-dependent evaluation strategy, because it fulfills the requirements of monotonicity according to Definition 3.2. In Figure 2 of Chapter 3 an exemplary search through the label space, which uses said evaluation strategy, is given. According to Definition 3.3 and Definition 3.1, monotonous evaluation functions cannot be decomposable, respectively anti-monotonous.

Disproof of Anti-Monotonicity: Equation 5.21 shows, how the recall of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated using example-based averaging and the rule-dependent evaluation strategy. According to the definition of example-based averaging, which is given in Equation 2.9, the overall performance of the rule is calculated by averaging the heuristic values, which are obtained for each example individually. The performance for an individual example evaluates to 0, if no true positives are covered. As predictions are only made for examples, which are covered by the rule's body B, this applies on all uncovered examples. Furthermore, if at least one true positive is covered, the performance for an example evaluates to 1. As a result, the performances, which are obtained for individual examples, may either be 0 or 1. This particularity becomes more obvious when taking a look at the rewritten form of the performance calculation, which is shown in the last line of Equation 5.21. It uses the Iverson bracket notation as introduced in Equation 2.18.

$$\begin{aligned} \delta_{rec,Mm,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{m} \cdot \sum_j \frac{\sum_{\hat{y}_i \in \hat{Y}} TP_i^j}{\sum_{\hat{y}_i \in \hat{Y}} P_i^j}, \text{ with } \begin{matrix} \sum_{\hat{y}_i \in \hat{Y}} P_i^j = \sum_{\hat{y}_i \in \hat{Y}} TP_i^j, \forall j(X_j \in C) \\ \sum_{\hat{y}_i \in \hat{Y}} TP_i^j = 0, \forall j(X_j \notin C) \end{matrix} \\ &\equiv \frac{1}{m} \cdot \sum_j \left[\sum_{\hat{y}_i \in \hat{Y}} TP_i^j > 0 \right] \end{aligned} \tag{5.21}$$

According to Equation 5.21, by adding additional label attributes to the head of a multi-label head rule, the overall performance cannot decrease. If the performance for an example was 0 before, adding the label attribute may either cause the performance to remain the same, or to become 1, if adding the label attribute causes a true positive to be covered. If the performance was 1 instead, it is guaranteed to remain the same, because at least one true positive was already covered before and will still be covered after adding the additional label attribute. Consequently, no heuristic value, which is obtained for an example, can decrease as the result of adding a label attribute to the head. Furthermore, because of the properties of the arithmetic mean (cf. Equation 5.6 and 5.7), the overall performance cannot decrease either. As a result, example-based recall meets the definition of monotonicity, which is given in Definition 3.2, if used together with the rule-dependent evaluation strategy. This implies Definition 3.3 and Definition 3.1 to not be met. ■

5.1.2.4 Macro-Averaging

When using macro-averaging, the performance of a multi-label head rule is calculated by obtaining heuristic values for each example and label at first. Afterwards, the overall performance is calculated by averaging the obtained values first example-wise and then label-wise, or vice versa. The first of both averaging orders is used to in the following in order to prove, that the recall metric is equivalent to the precision metric, when using macro-averaging and the rule-dependent evaluation strategy.

Proof of Decomposability: The calculation of the macro-averaged recall, when first averaging example-wise and then label-wise, is shown in Equation 5.22 below. Such as all considerations, which are made in this section, the calculation corresponds to using the rule-dependent evaluation strategy. According to Equation 5.22, the heuristic value, which is obtained for each example and label is either 0 or 1, depending on whether a true positive is covered, or not. A similar observation is made in Section 5.1.1.4, which deals with macro-averaged precision. In fact, as Equation 5.22 reveals, the formula for calculating the macro-averaged recall can be rewritten to match the calculation of the macro-averaged precision according to Equation 5.13. This shows, that both evaluation metrics are equivalent, if macro-averaging and the rule-dependent evaluation strategy is used.

$$
\begin{aligned}
\delta_{rec,MM,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \left(\frac{1}{m} \cdot \sum_j \frac{TP_i^j}{P_i^j} \right) \text{, with } \frac{TP_i^j}{P_i^j} = TP_i^j \text{ , } \forall i \forall j \\
&= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \sum_j \frac{TP_i^j}{m} \qquad \triangleright \text{ c.f (5.13), line 2} \qquad (5.22) \\
&\equiv \delta_{prec,MM}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{prec,Mm}(\hat{Y} \leftarrow B, T)
\end{aligned}
$$

As a result of Equation 5.22, the recall metric is proved to be decomposable according to Definition 3.3, when using macro-averaging and the rule-dependent evaluation strategy. This is due to the equivalence to its macro-averaged counterpart, which is shown to be decomposable in Section 5.1.1.4. Besides that, macro-averaged recall can be considered to be equivalent to example-based precision as well, because the macro-averaged and example-based variants of the precision metric are shown to be interchangeable, when using the rule-dependent evaluation, in that section as well. ■

5.1.3 Hamming Accuracy

Hamming accuracy, as introduced in Equation 2.14, calculates the percentage of correctly predicted relevant and irrelevant labels among all labels. Unlike the precision metric, it does also take true negatives and false negatives into account and therefore uncovered instances have an impact on the performance of a rule. In the following subsections, hamming accuracy is proved to be decomposable regarding the rule-dependent evaluation strategy, regardless of whether it is used together with micro-averaging, label-based averaging, example-based averaging or macro-averaging.

5.1.3.1 Micro-Averaging

In this section, micro-averaged hamming accuracy is examined in terms of anti-monotonicity, respectively decomposability. The proof, which is shown in the following, is similar to the one, that is used in Section 5.1.1.1 in order to prove the precision metric to be decomposable, when using micro-averaging together with the rule-dependent evaluation strategy. It is based on Equation 5.23, which shows how the micro-averaged hamming accuracy of a rule $\hat{y}_i \leftarrow B$, which contains a single label attribute $\hat{y}_i$ in its head, is calculated.

$$\begin{aligned}\delta_{hamm,mm,\not\perp}(\hat{y}_i \leftarrow B, T) &= \frac{\sum_j \left(TP_i^j + TN_i^j\right)}{\sum_j \left(P_i^j + N_i^j\right)} \text{ , with } \sum_j \left(P_i^j + N_i^j\right) = m \\ &= \frac{\sum_j \left(TP_i^j + TN_i^j\right)}{m}\end{aligned} \tag{5.23}$$

Proof of Decomposability: The proof, which is given in Equation 5.24 below, is based on showing, that Lemma 5.1 is met. For this reason, said equation, which denotes hows the micro-averaged hamming accuracy of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated, is rewritten in terms of averaging the performances of single-label head rules $\hat{y}_i \leftarrow B$ with $\hat{y}_i \in \hat{Y}$.

$$\begin{aligned}\delta_{hamm,mm,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j \left(TP_i^j + TN_i^j\right)}{\sum_{\hat{y}_i \in \hat{Y}} \sum_j \left(P_i^j + N_i^j\right)} \text{ , with } \sum_j \left(P_i^j + N_i^j\right) = m \text{ , } \forall i \\ &= \frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j \left(TP_i^j + TN_i^j\right)}{|\hat{Y}| \cdot m} \\ &= \frac{1}{|\hat{Y}|} \cdot \frac{\sum_{\hat{y}_i \in \hat{Y}} \sum_j \left(TP_i^j + TN_i^j\right)}{m} \\ &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \frac{\sum_j \left(TP_i^j + TN_i^j\right)}{m} \qquad \triangleright \text{ cf. (5.23), last line} \\ &\equiv \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \delta_{hamm,mm,\not\perp}(\hat{y}_i \leftarrow B, T)\end{aligned} \tag{5.24}$$

Rewriting Equation 5.24 to be in accordance with Lemma 5.1 is possible, because – as shown in Equation 5.23 – the performance for each predicted label calculates as the percentage of true positives and true negatives among all labels. The total number of labels equals the number of examples, which are available in the data set, and is denoted as m. As a result of Equation 5.24, which shows Lemma 5.1 to be met, micro-averaged hamming accuracy is implied to be decomposable according to Definition 3.3, when using the rule-dependent evaluation strategy. With respect to Lemma 3.1, it further follows, that said evaluation function can also be considered to be anti-monotonous according to Definition 3.1. ■

5.1.3.2 Label-based Averaging

When using label-based averaging together with the rule-dependent evaluation strategy, the used evaluation function is always decomposable. As mentioned earlier, this is due to that averaging strategy's definition, which is given in Equation 2.10 and which is in accordance with Lemma 5.1. The following proof does not only show the properties of decomposability to be met, but does also prove label-based averaging to be equivalent to micro-averaging, when using the hamming accuracy metric for a rule-dependent evaluation.

Proof of Decomposability: In Equation 5.25 the calculation of a multi-label head rule's performance, using label-based hamming accuracy and the rule-dependent evaluation strategy, is illustrated. According to the definition of said averaging strategy, a heuristic value is first calculated per relevant label at first. The obtained values are then averaged by using the arithmetic mean operation in order to obtain a single performance. As Equation 5.25 shows, when using label-based averaging, the heuristic value for an individual label is calculated in the same way as if micro-averaging was used. In Section 5.1.3.1 it is shown, that the micro-averaged hamming accuracy of a multi-label head rule $\hat{Y} \leftarrow B$ can be calculated by averaging the performances of single-label head rules $\hat{y}_i \leftarrow B$ with $\hat{y}_i \in \hat{Y}$. This does not only correspond to Lemma 5.1, but is also equal to the second line of Equation 5.25 below. As a result, the micro-averaged and label-based variants of the hamming accuracy can be considered to be equivalent.

$$\begin{aligned}
\delta_{hamm,mM,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \frac{\sum_j \left(TP_i^j + TN_i^j\right)}{\sum_j \left(P_i^j + N_i^j\right)} && \triangleright \text{ c.f (5.23), first line} \\
&\equiv \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \delta_{hamm,mm,\not\perp}(\hat{y}_i \leftarrow B, T) && \triangleright \text{ c.f (5.24), last line} \\
&\equiv \delta_{hamm,mm,\not\perp}(\hat{Y} \leftarrow B, T)
\end{aligned} \tag{5.25}$$

Equation 5.25 shows, that micro-averaged and label-based hamming accuracy are equivalent, when used for a rule-dependent evaluation. As a result, the label-based variant is implied to be decomposable according to Definition 3.3. This is a result of the proof, which is given in Section 5.1.3.1 in order to prove micro-averaged hamming accuracy to be decomposable. ■

5.1.3.3 Example-based Averaging

When using example-based averaging, according to Equation 2.9, for each example of the data set a heuristic value is calculated at first. By averaging all of these values using the arithmetic mean operation, a single performance is finally obtained. The following proof shows, that the example-based variant of the hamming accuracy metric is not only decomposable, when using the rule-dependent evaluation strategy, but that it is also equivalent to its micro-averaged counterpart.

Proof of Decomposability: Equation 5.26 shows, how the example-based hamming accuracy of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated, when using a rule-dependent evaluation. According to the definition of example-based averaging, the evaluation function is applied to each example individually. According to Equation 2.14, hamming accuracy measures the percentage of true positives and true negatives among all labels. When using the rule-dependent evaluation strategy, the total number of relevant labels per example equals the number labels, which are contained in the head $\hat{Y}$. As a result, Equation 5.26 can be rewritten by using the term $|\hat{Y}|$ for denoting the number of labels. By further exploiting the first-order homogeneity of the arithmetic mean operation, the equation can be converted into a form, which is equal to the fourth line of Equation 5.24. Consequently, example-based averaging is proved to be equivalent to its micro-averaged counterpart, when using the rule-dependent evaluation strategy.

$$
\begin{aligned}
\delta_{hamm,Mm,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{m} \cdot \sum_j \frac{\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j\right)}{\sum_{\hat{y}_i \in \hat{Y}} \left(P_i^j + N_i^j\right)} \text{ , with } \sum_{\hat{y}_i \in \hat{Y}} \left(P_i^j + N_i^j\right) = |\hat{Y}| \text{ , } \forall j \\
&= \frac{1}{m} \sum_j \frac{\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j\right)}{|\hat{Y}|} \\
&= \frac{1}{m} \sum_{\hat{y}_i \in \hat{Y}} \sum_j \frac{TP_i^j + TN_i^j}{|\hat{Y}|} \\
&= \frac{1}{|\hat{Y}|} \sum_{\hat{y}_i \in \hat{Y}} \sum_j \frac{TP_i^j + TN_i^j}{m} \\
&= \frac{1}{|\hat{Y}|} \sum_{\hat{y}_i \in \hat{Y}} \frac{\sum_j \left(TP_i^j + TN_i^j\right)}{m} \qquad \triangleright \text{ c.f (5.24), line 4} \\
&\equiv \delta_{hamm,mm,\not\perp}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{hamm,mM,\not\perp}(\hat{Y} \leftarrow B, T)
\end{aligned}
\tag{5.26}
$$

From the equivalence, which is shown in Equation 5.26, two conclusions follow: On the one hand, example-based averaging can not only be considered to be equivalent to micro-averaged hamming accuracy, when using the rule-dependent evaluation strategy, but it is also implied to be equivalent to the label-based variant. This is, because the use of label-based averaging is shown to be equivalent to using micro-averaging in Section 5.1.3.2. On the other hand, example-based hamming accuracy is implied to meet the properties of decomposable evaluation functions. This implication is based on Section 5.1.3.1 in which it is shown, that micro-averaged hamming accuracy fulfills Definition 3.3. ■

5.1.3.4 Macro-Averaging

According to Equation 2.11, when using macro-averaging, the evaluation function is first applied to each example and label individually. In order to obtain a single heuristic value, the performances, which are calculated in that way, are finally averaged by using the arithmetic mean operation first example-wise

and then label-wise, or vice versa. The first of both averaging orders is used in the following in order to prove macro-averaging to be equivalent to micro-averaging, when used together with the hamming accuracy metric and the rule-dependent evaluation strategy.

Proof of Decomposability: Hamming accuracy measures the percentage of correctly predicted relevant and irrelevant labels among all labels. When using macro-averaging, the evaluation function is applied to each example and label individually. As it is shown in Equation 5.27 below, the total number of labels evaluates to 1 in such case. By exploiting the first-order homogeneity of the arithmetic mean operation, the equation can be rewritten as shown in the third line of Equation 5.27. The rewritten form of the equation corresponds to the fourth line of Equation 5.24. As a result, macro-averaged hamming accuracy and its micro-averaged counterpart are shown to be equivalent, when used together with the rule-dependent evaluation strategy.

$$
\begin{aligned}
\delta_{hamm,MM,\not\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \left(\frac{1}{m} \cdot \sum_j \frac{TP_i^j + TN_i^j}{P_i^j + N_i^j} \right) \text{, with } P_i^j + N_i^j = 1 \text{ , } \forall i \forall j \\
&= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \left(\frac{1}{m} \cdot \sum_j TP_i^j + TN_i^j \right) \\
&= \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \frac{\sum_j \left(TP_i^j + TN_i^j \right)}{m} \qquad \triangleright \text{ c.f (5.24), line 4} \\
&\equiv \delta_{hamm,mm,\not\perp}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{hamm,mM,\not\perp}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{hamm,Mm,\not\perp}(\hat{Y} \leftarrow B, T)
\end{aligned}
\tag{5.27}
$$

As a result of Equation 5.27, which proves macro-averaged hamming accuracy to be equivalent to micro-averaged hamming accuracy, when using the rule-dependent evaluation strategy, the macro-averaged variant is implied to be decomposable according to Definition 3.3. This is, because micro-averaged hamming accuracy is shown to be decomposable in Section 5.1.3.1. Furthermore, as the use of label-based, as well as example-based averaging is shown to be equivalent to using micro-averaging in Section 5.1.3.2, respectively in Section 5.1.3.3, it follows that macro-averaged hamming accuracy is equivalent to both of these variants as well. As a conclusion it can be stated, that all averaging strategies – namely micro-averaging, label-based averaging, example-based averaging and macro-averaging – are equivalent when used together with the hamming accuracy metric and the rule-dependent evaluation strategy. ■

5.1.4 F-Measure

According to Equation 2.16, the F-Measure is defined as the weighted harmonic mean of precision and recall, where the weight of the recall is β^2-times the weight of the precision. If $\beta = 0$, the F-Measure is equal to the precision metric and therefore the considerations in this section focus on scenarios where $\beta > 0$. In general, the harmonic mean of two positive real numbers x_1 and x_2 with weights w_1 and w_2 is defined as follows:

$$
H(x_1, x_2) := \frac{w_1 + w_2}{\frac{w_1}{x_1} + \frac{w_2}{x_2}} \tag{5.28}
$$

Harmonic mean

Such as the arithmetic mean, the harmonic mean is one of the Pythagorean means and therefore both operations share some common properties [Heath, 1921]. If $w_1, w_2 > 0$, the weighted harmonic mean strictly lies between the values it is calculated from. In the remainder of this work, this is referred to as the "averaging" property of the harmonic mean:

$$min(x_1, x_2) < H(x_1, x_2) < max(x_1, x_2) \tag{5.29}$$

Averaging

Furthermore, if the harmonic mean is calculated from two equal values, its "value preservation" property applies. Said property states, that the harmonic mean of two identical values equals the given values, regardless of their weights:

$$H(x_1, x_2) = x_1 = x_2 \text{ , with } x_1 = x_2 \tag{5.30}$$

Value preservation

The F-Measure can be rewritten in terms of the harmonic mean function H, where the recall has a weight of β^2 and the precision has a weight of 1 (cf. 2.16). As a result, if $\beta > 0$, the "averaging" and "value preservation" properties, which are introduced in Equation 5.29, respectively Equation 5.30, also apply to the F-Measure.

5.1.4.1 Micro-Averaging

The following proof shows, that the micro-averaged F-Measure is decomposable, when using the rule-dependent evaluation strategy. The proof is based on micro-averaged recall and precision being proved to be decomposable in Section 5.1.2.1, respectively in Section 5.1.1.1, when using said evaluation strategy. As multiple evaluation functions take part in the proof, different notations are necessary to distinguish between the best possible performance according to different evaluation functions. For example, the best possible performance according to the F-Measure is denoted by using the following syntax:

$$h^F_{max} \tag{5.31}$$

Furthermore, the following proof is based on the fact, that the best possible performance according to the F-Measure can not be greater than the maximum of the best performances according to recall and precision. This inequality is denotated by Equation 5.32 below:

$$h^F_{max} \leq max\left(h^{rec}_{max}, h^{prec}_{max}\right) \tag{5.32}$$

Proof of Decomposability: Equation 5.34 proves the first property of decomposability, which is given in Definition 3.3 i), to be met. By rewriting the F-Measure in terms of the harmonic mean operation H, the equation is converted to solely depend on micro-averaged recall and precision. Both of these metrics are shown to fulfill the property, which is given in Definition 3.3 i), in Section 5.1.2.1, respectively in Section 5.1.1.1. Furthermore, as the premise of the proof, it is assumed, that the best possible performance according to the recall metric is greater or equal than the best possible performance according to the precision metric:

$$h^{rec}_{max} \geq h^{prec}_{max} \tag{5.33}$$

This assumption can be made without loss of generality, because the Equation 5.34 can easily be adapted to the opposite, i.e. to a premise, where the best possible performance according to the precision metric is greater or equal than the best possible performance according to the recall metric.

$$
\begin{aligned}
&\exists i \left(\hat{y}_i \in \hat{Y} \wedge \delta_{F,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) < h^F_{max} \leq h^{rec}_{max} \right) \\
\equiv &\exists i \left(\hat{y}_i \in \hat{Y} \wedge H\left(\delta_{rec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T), \delta_{prec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) \right) < h^{rec}_{max} \right) \\
\xRightarrow[\text{and (5.29)}]{\text{w.r.t. (5.30)}} &\exists i \left(\hat{y}_i \in \hat{Y} \wedge \left(\delta_{rec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) < h^{rec}_{max} \wedge \delta_{prec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) < h^{rec}_{max} \right) \right. \\
&\left. \vee \left(\delta_{prec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) < h^{rec}_{max} \wedge \delta_{rec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) \leq h^{rec}_{max} \right) \right) \\
\xRightarrow[\delta_{rec,mm,\not\parallel} \text{ and } \delta_{prec,mm,\not\parallel}]{\text{w.r.t. decomposability of}} &\left(\delta_{rec,mm,\not\parallel}(\hat{Y} \leftarrow B, T) < h^{rec}_{max} \wedge \delta_{prec,mm,\not\parallel}(\hat{Y} \leftarrow B, T) < h^{rec}_{max} \right) \\
&\vee \left(\delta_{prec,mm,\not\parallel}(\hat{Y} \leftarrow B, T) < h^{rec}_{max} \wedge \delta_{rec,mm,\not\parallel}(\hat{Y} \leftarrow B, T) \leq h^{rec}_{max} \right) \\
\xRightarrow[\text{and (5.29)}]{\text{w.r.t. (5.30)}} &H\left(\delta_{rec,mm,\not\parallel}(\hat{Y} \leftarrow B, T), \delta_{prec,mm,\not\parallel}(\hat{Y} \leftarrow B, T) \right) < h^F_{max} \leq h^{rec}_{max} \\
\equiv &\delta_{F,mm,\not\parallel}(\hat{Y} \leftarrow B, T) < h^F_{max}
\end{aligned}
\tag{5.34}
$$

In Equation 5.34 is assumed, that the F-Measure of a single-label head rule $\hat{y}_i \leftarrow B$ is less then the best possible performance (cf. Equation 5.34, line 1 and 2). This is in accordance with the equation, which is given in Definition 3.3 i). Note, that –according to the premise of the proof – the best possible performance h^F_{max} cannot be greater than h^{rec}_{max}. Because the F-Measure can be rewritten as the harmonic mean H of precision and recall, it follows from the averaging and value preservation properties of the harmonic mean operation (cf. Equation 5.29 and Equation 5.30), that either the recall or the precision of the single-label rule $\hat{y}_i \leftarrow B$ must be less than the best possible performance h^F_{max}, respectively h^{rec}_{max}. Due to the inequality, which is given in Equation 5.33, h^{rec}_{max} can be considered as an upper limit for both recall, as well as precision (cf. Equation 5.34, line 3). Furthermore, from the decomposability of the precision and recall metric, it follows, that a multi-label head rule $\hat{Y} \leftarrow B$, which contains the label attribute $\hat{y}_i$ in its head, cannot outperform the best possible performance h^F_{max} (cf. Equation 5.34, line 5, 7 and 8). In order to prove the second property of decomposability to be met, Equation 5.35 uses a similar approach. However, it is not based on the premise, which is given in Equation 5.33.

$$
\begin{aligned}
&\delta_{F,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) = h^F_{max}\ ,\ \forall \hat{y}_i \left(\hat{y}_i \in \hat{Y} \right) \\
\equiv &H\left(\delta_{rec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T), \delta_{prec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) \right) = h^F_{max}\ ,\ \forall \hat{y}_i \left(\hat{y}_i \in \hat{Y} \right) \\
\xRightarrow[\text{and (5.29)}]{\text{w.r.t. (5.30)}} &\delta_{rec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) = \delta_{prec,mm,\not\parallel}(\hat{y}_i \leftarrow B, T) = h^F_{max}\ ,\ \forall \hat{y}_i \left(\hat{y}_i \in \hat{Y} \right) \\
\xRightarrow[\delta_{rec,mm,\not\parallel} \text{ and } \delta_{prec,mm,\not\parallel}]{\text{w.r.t. decomposability of}} &\delta_{rec,mm,\not\parallel}(\hat{Y} \leftarrow B, T) = \delta_{prec,mm,\not\parallel}(\hat{Y} \leftarrow B, T) = h^F_{max} \\
\xRightarrow[\text{and (5.29)}]{\text{w.r.t. (5.30)}} &H\left(\delta_{rec,mm,\not\parallel}(\hat{Y} \leftarrow B, T), \delta_{prec,mm,\not\parallel}(\hat{Y} \leftarrow B, T) \right) = h^F_{max} \\
\equiv &\delta_{F,mm,\not\parallel}(\hat{Y} \leftarrow B, T) = h^F_{max}
\end{aligned}
\tag{5.35}
$$

From Equation 5.34 and Equation 5.35 follows, that the F-Measure meets the properties of decomposable evaluation functions according to Definition 3.3, when using micro-averaging together with the rule-dependent evaluation strategy. Although the given proof does only apply if $\beta > 0$, the evaluation function can be considered to be decomposable regardless of the β-parameter's value. This is, because the F-Measure is equivalent to the precision metric, if $\beta = 0$. Precision is shown to be decomposable, regarding micro-averaging and the rule-dependent evaluation strategy, in Section 5.1.1.1.

■

5.1.4.2 Label-based Averaging

Proof of Decomposability: As already mentioned, when using label-based averaging for measuring the rule-dependent performance of multi-label head rules, the used evaluation function is always decomposable. This does also apply, when using the label-based F-Measure. In Equation 5.36 the calculation of multi-label head rule's performance according to said evaluation function and using the rule-dependent evaluation strategy is illustrated. As it can be seen, a heuristic value is obtain for each relevant label at first. The overall performance of the given rule finally results from averaging all obtained values using the arithmetic mean operation.

$$\delta_{F,mM,\not\perp}(\hat{Y} \leftarrow B, T) = \frac{1}{n} \cdot \sum_{\hat{y}_i \in \hat{Y}} \delta_{F,mM,\not\perp}(\hat{y}_i \leftarrow B, T) \tag{5.36}$$

The calculation, which is shown in Equation 5.36, corresponds to the equation, which is part of Lemma 5.1. As the fulfillment of Lemma 5.1 is sufficient for Definition 3.3 to be met, the label-based variant of the F-Measure can be considered to be decomposable, if the rule-dependent evaluation strategy is used. The given proof does apply, regardless of which β-parameter is used to trade-off between precision and recall.

■

5.1.4.3 Example-based Averaging

In this section, the F-Measure is examined in terms of anti-monotonicity, respectively decomposability, when using example-based averaging together with the rule-dependent evaluation strategy. The proof, which is given in the following, is based on rewriting the calculation of a multi-label head rule's performance according to Equation 5.37. The equation uses the (weighted) harmonic mean operation H in order to denote the F-Measure of an individual example of a training data set. It is further based on the observation, that recall and precision both evaluate to 0, if no true positives are covered for an example. If at least one true positive is covered, the recall always evaluates to 1, instead. In such case, the F-Measure, which is obtained for an individual example, can be denoted as the harmonic mean of 1 and the heuristic value, which is calculated by using the precision metric. Because the recall is constant, for examples for which any true positives are covered, the F-Measure solely depends on the precision metric in such case.

$$\delta_{F,Mm,\not\perp}(\hat{Y} \leftarrow B, T) = \frac{1}{m} \cdot \sum_{j} \begin{cases} H\left(1, \frac{\sum_{\hat{y}_i \in \hat{Y}} TP_i^j}{\sum_{\hat{y}_i \in \hat{Y}} p_i^j}\right), & \text{if } \sum_{\hat{y}_i \in \hat{Y}} TP_i^j > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.37}$$

Proof of Decomposability: In Equation 5.38 and Equation 5.37, the F-Measure is proved to meet the properties of decomposability, when using example-based averaging and a rule-dependent evaluation. In Equation 5.38, the first property of decomposable evaluation functions, which is given in Definition 3.3

i), is shown to be met. In Equation 5.39, the property which corresponds to Definition 3.3 ii), is proved to be fulfilled accordingly. Both equations are based on the observation, that the example-based F-Measure of a rule primarily depends on the precision metric, as revealed by Equation 5.37. Consequently, if the F-Measure of a single-label head rule $\hat{y}_i \leftarrow B$ is less than the best possible performance h_{max}, it follows from the averaging and value preservation properties of the harmonic mean (cf. Equation 5.29 and Equation 5.30), that the precision of that rule is less than h_{max} as well (cf. Equation 5.38, line 1 and 2). The example-based variant of the precision metric is shown to be decomposable in Section 5.1.1.3, when using the rule-dependent evaluation strategy. Therefore it is implied, that the precision of a multi-label head rule $\hat{Y} \leftarrow B$ is less than h_{max}, if one of the corresponding single-label head rules with heads $\hat{y}_i \in \hat{Y}$ does not reach that performance (cf. Equation 5.38, line 3). As the F-Measure calculates as the harmonic mean of recall and precision and because the recall is always 1 – respectively 0, if the precision is 0 as well – from the averaging and value preservation properties of the harmonic mean follows, that the F-Measure of a multi-label head rule $\hat{Y} \leftarrow B$ is less than h_{max} in such case as well (cf. Equation 5.38, line 5 and 6).

$$\exists i \left(\hat{y}_i \in \hat{Y} \wedge \delta_{F,Mm,\not\perp}(\hat{y}_i \leftarrow B, T) < h_{max} \right)$$

$$\xRightarrow[\text{and (5.37)}]{\text{w.r.t. (5.30), (5.29)}} \exists i \left(\hat{y}_i \in \hat{Y} \wedge \delta_{prec,Mm,\not\perp}(\hat{y}_i \leftarrow B, T) < h_{max} \right)$$

$$\xRightarrow[\delta_{prec,Mm,\not\perp}]{\text{w.r.t. decomposability of}} \delta_{prec,Mm,\not\perp}(\hat{Y} \leftarrow B, T) < h_{max} \tag{5.38}$$

$$\xRightarrow[\text{and (5.37)}]{\text{w.r.t. (5.30), (5.29)}} H\left(\delta_{rec,Mm,\not\perp}(\hat{Y} \leftarrow B, T), \delta_{prec,Mm,\not\perp}(\hat{Y} \leftarrow B, T) \right) < h_{max}$$

$$\equiv \delta_{F,Mm,\not\perp}(\hat{Y} \leftarrow B, T) < h_{max}$$

In order to prove the second property of decomposability, which is given in Definition 3.3 ii), to be met by the example-based F-Measure, Equation 5.39 is used. It is based on similar implications as discussed above.

$$\delta_{F,Mm,\not\perp}(\hat{y}_i \leftarrow B, T) = h_{max} \;,\; \forall \hat{y}_i \left(\hat{y}_i \in \hat{Y} \right)$$

$$\xRightarrow[\text{and (5.37)}]{\text{w.r.t. (5.30), (5.29)}} \delta_{prec,Mm,\not\perp}(\hat{y}_i \leftarrow B, T) = h_{max} \;,\; \forall \hat{y}_i \left(\hat{y}_i \in \hat{Y} \right)$$

$$\xRightarrow[\delta_{prec,Mm,\not\perp}]{\text{w.r.t. decomposability of}} \delta_{prec,Mm,\not\perp}(\hat{Y} \leftarrow B, T) = h_{max} \tag{5.39}$$

$$\xRightarrow[\text{and (5.37)}]{\text{w.r.t. (5.30), (5.29)}} H\left(\delta_{rec,Mm,\not\perp}(\hat{Y} \leftarrow B, T), \delta_{prec,Mm,\not\perp}(\hat{Y} \leftarrow B, T) \right) = h_{max}$$

$$\equiv \delta_{F,Mm,\not\perp}(\hat{Y} \leftarrow B, T) = h_{max}$$

Equation 5.38 and Equation 5.39 prove, that the properties of decomposability, according to Definition 3.3, are met, when using example-based F-Measure together with the rule-dependent evaluation strategy for measuring the performance of multi-label head rules. The proof, which is given in this section, is independent of the used β-parameter. ■

5.1.4.4 Macro-Averaging

When using macro-averaging for measuring the performance of a multi-label head rule, a heuristic value is obtained per example and label beforehand. By averaging the obtained values first example-wise and then label-wise, or vice versa, a single performance can be calculated afterwards. Based on the first of both averaging orders, the following proof shows, that the macro-averaged F-Measure is equivalent to macro-averaged recall and precision, when using the rule-dependent evaluation strategy.

Proof of Decomposability: Equation 5.40 shows, how the macro-averaged F-Measure of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated, when using the rule-dependent evaluation strategy. The equation is written in terms of the (weighted) harmonic mean operation H, which is used to trade-off between recall and precision.

$$
\begin{aligned}
\delta_{F,MM,\not\parallel}(\hat{Y} \leftarrow B, T) =& \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \left(\frac{1}{m} \cdot \sum_j H\left(\frac{TP_i^j}{P_i^j}, \frac{TP_i^j}{p_i^j} \right) \right) \text{, with } \frac{TP_i^j}{P_i^j} = \frac{TP_i^j}{p_i^j} = TP_i^j \text{ , } \forall i \forall j \\
=& \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \left(\frac{1}{m} \cdot \sum_j H\left(TP_i^j, TP_i^j \right) \right) && \triangleright \text{ (5.30) applies} \\
=& \frac{1}{|\hat{Y}|} \cdot \sum_{\hat{y}_i \in \hat{Y}} \sum_j \frac{TP_i^j}{m} && \triangleright \text{ c.f (5.13), line 2 and (5.22), line 2} \\
\equiv& \delta_{rec,MM,\not\parallel}(\hat{Y} \leftarrow B, T) \\
\equiv& \delta_{prec,MM,\not\parallel}(\hat{Y} \leftarrow B, T) \\
\equiv& \delta_{prec,Mm,\not\parallel}(\hat{Y} \leftarrow B, T)
\end{aligned}
\tag{5.40}
$$

As shown in Equation 5.40 above, for each example and label, recall and precision both evaluate to either 0 or 1, depending on whether a true positive is covered, or not. This particularity corresponds to observations, which are made in Section 5.1.2.4 and Section 5.1.1.4. As a result, due to its value preservation property (cf. Equation 5.30), the harmonic mean – and consequently the F-Measure – of the heuristic values, which are obtained for each example and label, is always identical to recall and precision. This implies, that the F-Measure is equivalent to both, the recall and precision metric, when using macro-averaging together with the rule-dependent evaluation strategy. Due to this equivalence, the macro-averaged F-Measure is proven to be decomposable according to Definition 3.3, if the rule-dependent evaluation strategy is used. This is, because the macro-averaged variants of the recall and precision metrics are shown to fulfill said definition in Section 5.1.2.4, respectively Section 5.1.1.4. Furthermore, macro-averaged precision is shown to be equivalent to example-based precision in Section 5.1.1.4. As a result, the macro-averaged F-Measure is implied to be equivalent to that evaluation strategy as well. The equivalences, which are shown in Equation 5.40, are independent of the β-parameter's value. ■

5.1.5 Subset Accuracy

According to the definition of subset accuracy, which is given in Equation 2.18, this evaluation metric measures the percentage of perfectly predicted label vectors among all examples of a data set. When using the rule-dependent evaluation strategy, a label vector is considered to be predicted perfectly, if all

labels, which are set by a rule, are predicted correctly. The labels, which are not predicted by a rule, are not taken into account in such case. As already mentioned in Section 2.3.3, the subset accuracy metric is only defined in terms of using example-based averaging. Therefore the remaining averaging strategies, which are discussed in Section 2.3.2, must not be considered at this point. In the following proof, which shows, that subset accuracy meets the properties of anti-monotonicity, the evaluation function is written in terms of true positives and true negatives as shown in Equation 5.41 below. According to said equation, the performance for an individual example evaluates to 1, if the number of correctly predicted labels (true positives and true negatives) equals the number of labels, which are predicted by a rule. If the sum of true positives and false negatives is less than the number of predicted labels, at least one label is predicted incorrectly and the performance therefore evaluates to 0.

$$\delta_{acc,\not\perp}(\hat{Y} \leftarrow B, T) = \frac{1}{m} \cdot \sum_j \left[\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j \right) = |\hat{Y}| \right] \tag{5.41}$$

Proof of Decomposability: The proof, which is shown below, proves the properties of anti-monotonicity, according to Definition 3.1, to be met by the subset accuracy metric, when using the rule-dependent evaluation strategy. In Equation 5.42 two multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ take part. In accordance with the equation, which is given in Definition 3.1, both rules share a common body B and the head $\hat{Y}_s$ is assumed to contain additional label attributes besides those of $\hat{Y}_p$. Furthermore, it is assumed, that the rule $\hat{Y}_p \leftarrow B$ outperforms the rule $\hat{Y}_s \leftarrow B$.

$$\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta_{acc,\not\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{acc,\not\perp}(\hat{Y}_p \leftarrow B, T) \\
\Longrightarrow &\frac{1}{m} \cdot \sum_j \left[\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j \right) = |\hat{Y}| \right] \Bigg|_{\hat{Y}_s \leftarrow B,T} < \frac{1}{m} \cdot \sum_j \left[\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j \right) = |\hat{Y}| \right] \Bigg|_{\hat{Y}_p \leftarrow B,T} \leq h_{max} \\
\Longrightarrow &\exists j \left(0 = \left[\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j \right) = |\hat{Y}| \right] \Bigg|_{\hat{Y}_s \leftarrow B,T} < \left[\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j \right) = |\hat{Y}| \right] \Bigg|_{\hat{Y}_p \leftarrow B,T} = 1 \right) \\
\Longrightarrow &\exists \hat{y}_i \exists j \left(\hat{y}_i \in \hat{Y}_s \wedge \left(TP_i^j + TN_i^j \right) < |\hat{Y}| \Big|_{\hat{Y}_s \leftarrow B,T} \right) \\
\Longrightarrow &\exists \hat{y}_i \exists j \left(\hat{y}_i \in \hat{Y}_a \wedge \left(TP_i^j + TN_i^j \right) < |\hat{Y}| \Big|_{\hat{Y}_a \leftarrow B,T} \right), \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
\Longrightarrow &\exists j \left(\left[\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j \right) = |\hat{Y}| \right] \Bigg|_{\hat{Y}_a \leftarrow B,T} = 0 \right), \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
\Longrightarrow &\frac{1}{m} \cdot \sum_j \left[\sum_{\hat{y}_i \in \hat{Y}} \left(TP_i^j + TN_i^j \right) = |\hat{Y}| \right] \Bigg|_{\hat{Y}_a \leftarrow B,T} < h_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
\equiv &\delta_{acc,\not\perp}(\hat{Y}_a \leftarrow B, T) < h_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right)
\end{aligned} \tag{5.42}$$

In Equation 5.42 it is concluded, that when using the rule $\hat{Y}_s \leftarrow B$, the performance for at least one example is less, than when using the rule $\hat{Y}_p \leftarrow B$ (cf. Equation 5.42, third line). As the performance for an individual example may either be 0 or 1, the performance for said example must evaluate to 0, when evaluated against the rule $\hat{Y}_s \leftarrow B$, respectively to 1, when considering the rule $\hat{Y}_p \leftarrow B$. Because the performance only becomes 0, if the number of correctly predicted labels is less than the number of predicted labels, this leads to the conclusion, that the head $\hat{Y}_p$ contains at least one label attribute $\hat{y}_i$, which predicts the corresponding label λ_i of the respective example incorrectly. By adding additional label attributes, the label λ_1 will still be predicted incorrectly. Therefore, for all multi-label head rules $\hat{Y}_a \leftarrow B$, which result from adding additional label attributes to the head $\hat{Y}_s$, the performance of the respective example evaluates to 0. This implies, that by adding additional label attributes to the rule $\hat{Y}_s \leftarrow B$, neither the performance of the rule $\hat{Y}_p \leftarrow B$, nor the best possible performance h_{max}, can be reached. As this is in accordance with Definition 3.1, the properties of anti-monotonicity are shown to be met by the subset accuracy metric, when using the rule-dependent evaluation strategy. ■

5.2 Rule-independent Evaluation

In this section, the evaluation functions, which are given in Section 2.3.3, are examined in terms of anti-monotonicity and decomposability, when using the rule-independent evaluation strategy. As discussed in Section 3.1, all labels, regardless of whether they are predicted by a rule, or not, must be taken into account by a rule-independent evaluation. If a label is not set by a rule, it is assumed to be predicted as irrelevant. In order to simplify the notation of the proofs, which are shown throughout this section, the following functions $r, T \rightarrow \mathbb{N}$ are used. The function TP^i_{max} returns the maximum number of true positives a rule r may cover, regarding the label λ_i of a data set T. The return value of that function solely depends on the examples, which are covered by the rule's body. It does not take the rule's head into account, but returns the maximum number of true positives the best possible head could reach.

$$TP^i_{max}(r, T) := max\left(\sum_{(X_j, Y_j) \in C} \left[y_i \in Y_j = 1\right], \sum_{(X_j, Y_j) \in C} \left[y_i \in Y_j = 0\right]\right) \tag{5.43}$$

The function TP_{max} is similar to the function, which is given above. It returns the maximum number of true positives, which can be reached by a rule r on a data set T. Unlike the function TP^i_{max}, which does only consider a single label, the return value of the function TP_{max} depends on all available labels.

$$TP_{max}(r, T) := \sum_i TP^i_{max} \tag{5.44}$$

In addition to the functions, which are shown above, the variable h_{max} is used to denote the best possible performance, which can be reached on a data set, using a specific evaluation function and averaging strategy. This corresponds to the semantic of the variable h_{max} as used in Definition 3.1 and Definition 3.3. In the following proofs, the variable h_{max} is used as a "symbolic" value. The actual value, which depends on the used evaluation function and averaging strategy, is usually not given. Moreover, in the context of label-based averaging, the variable h^i_{max} is used to denote the best possible performance, which can be reached for an individual label λ_i.

5.2.1 Precision

In this section, the precision metric is examined in terms of anti-monotonicity, when used together with the rule-independent evaluation strategy. As shown in the following subsections, said evaluation function meets the properties of anti-monotonicity, regardless of whether micro-averaging, label-based averaging, example-based averaging or macro-averaging is used.

5.2.1.1 Micro-Averaging

Equation 5.45 illustrates, how the micro-averaged precision of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated, when using the rule-independent evaluation strategy. According to Equation 2.12, precision measures the percentage of true positives among the labels of all covered examples. Whereas the number of covered examples is denoted as $|C|$, the number of labels corresponds to the variable n.

$$\begin{aligned} \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) &= \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j p_i^j} \text{ , with } \sum_i \sum_j p_i^j = n \cdot |C| \\ &= \frac{\sum_i \sum_j TP_i^j}{n \cdot |C|} \end{aligned} \tag{5.45}$$

Proof of Anti-Monotonicity: In Equation 5.46, the precision metric is shown to meet the properties of anti-monotonicity, when using the rule-independent evaluation strategy. In accordance with Definition 3.1, which formally specifies the properties of anti-monotonicity, two multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ take part in the equation.

$$\begin{aligned} & \hat{Y}_p \subset \hat{Y}_s \wedge \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\ \Longrightarrow & \left. \frac{\sum_i \sum_j TP_i^j}{n \cdot |C|} \right|_{\hat{Y}_s \leftarrow B,T} < \left. \frac{\sum_i \sum_j TP_i^j}{n \cdot |C|} \right|_{\hat{Y}_p \leftarrow B,T} \leq h_{max} \\ \Longrightarrow & \left. \sum_i \sum_j TP_i^j \right|_{\hat{Y}_s \leftarrow B,T} < \left. \sum_i \sum_j TP_i^j \right|_{\hat{Y}_p \leftarrow B} \leq TP_{max} \\ \Longrightarrow & \exists i \left(\left. \sum_j TP_i^j \right|_{\hat{Y}_s \leftarrow B,T} < TP_{max}^i \right) \\ \Longrightarrow & \exists i \left(\left. \sum_j TP_i^j \right|_{\hat{Y}_a \leftarrow B,T} < TP_{max}^i \right) , \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\ \Longrightarrow & \left. \sum_i \sum_j TP_i^j \right|_{\hat{Y}_a \leftarrow B,T} < TP_{max} , \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\ \Longrightarrow & \left. \frac{\sum_i \sum_j TP_i^j}{n \cdot |C|} \right|_{\hat{Y}_a \leftarrow B,T} < h_{max} , \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\ \equiv & \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h_{max} , \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \end{aligned} \tag{5.46}$$

Whereas the rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$, which take part in Equation 5.46, share a common body B, their heads differ. The head $\hat{Y}_s$ is assumed to contain additional label attributes besides those of the head $\hat{Y}_p$. Moreover – in accordance with Definition 3.1 –, the rule $\hat{Y}_p \leftarrow B$ is assumed to reach a higher performance than the rule $\hat{Y}_s \leftarrow B$. In the second line of Equation 5.46, the performance calculations regarding the rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ are rewritten with respect to Equation 5.45. As both of the resulting fractions share a common denominator $n \cdot |C|$, it is implied, that the difference between both rules' performances exclusively results from the true positives they cover. Because the performance of the rule $\hat{Y}_s \leftarrow B$ is assumed to be less than the performance of the rule $\hat{Y}_p \leftarrow B$, the first rule is implied to cover less true positives than the latter. As a consequence, the rule $\hat{Y}_s \leftarrow B$ cannot reach the best possible performance h_{max} (cf. Equation 5.46, line 3). If the maximum number of true positives is not reached by a rule, this is, because for at least one label less true positives than possible are covered – i.e. for a label λ_i the rule $\hat{Y}_s \leftarrow B$ does not reach TP^i_{max} (cf. Equation 5.46, line 4). By adding additional label attributes to the head $\hat{Y}_s$, the maximum number of true positives TP_{max} cannot be covered either, because the label for which TP^i_{max} is not reached, is still contained in the head (cf. Equation 5.46, line 5 and 6). As a result, no rule $\hat{Y}_a \leftarrow B$, which results from adding additional label attributes to the head of the rule $\hat{Y}_s \leftarrow B$, are able to reach the best possible heuristic value h_{max}. As this is in accordance with Definition 3.1, the definition of anti-monotonicity is met. Consequently, the precision metric is shown to be anti-monotonous, when used together with micro-averaging and a rule-independent evaluation. ■

5.2.1.2 Label-based Averaging

Proof of Anti-Monotonicity: Equation 5.47, which is shown below, proves the use of label-based averaging to be equivalent to the use of micro-averaging, regarding the precision metric and a rule-independent evaluation.

$$\begin{aligned}
\delta_{prec,mM,\amalg}(\hat{Y} \leftarrow B, T) &= \frac{1}{n} \cdot \sum_i \frac{\sum_j TP_i^j}{\sum_j p_i^j} \text{ , with } \sum_j p_i^j = |C| \text{ , } \forall i \\
&= \frac{\sum_i \sum_j TP_i^j}{n \cdot |C|} \qquad \triangleright \text{ c.f (5.45), last line} \\
&\equiv \delta_{prec,mm,\amalg}(\hat{Y} \leftarrow B, T)
\end{aligned} \tag{5.47}$$

As Equation 5.47 illustrates, when using label-based averaging, a heuristic value is calculated for each available label. By averaging the obtained values, a single performance can be computed afterwards. The heuristic value, which is obtained for each label, calculates as the percentage of true positives among all covered examples $|C|$. This particularity can be exploited in order to rewrite the calculation to be in accordance with Equation 5.45. As Equation 5.45 corresponds to the use of micro-averaging, it follows, that label-based precision is equivalent to micro-averaged precision, when using the rule-independent evaluation strategy. Moreover, as the latter variant is shown to be anti-monotonous in Section 5.2.1.1, label-based precision is implied to meet the definition of anti-monotonicity as well. ■

5.2.1.3 Example-based Averaging

The proof, which is given in the following, in order to prove example-based precision to be anti-monotonous, when using the rule-independent evaluation strategy, is similar to the one, which is given

in Section 5.2.1.1 above. It is based on Equation 5.48, which shows, how the performance of a multi-label rule $\hat{Y} \leftarrow B$ is calculated. According to the definition of example-based averaging, which is given in Equation 2.9, heuristic values are obtained for each example at first. According to the precision metric, these values measure the number of true positives among all labels n. The overall performance of a rule finally results from the arithmetic mean of the heuristic values, which have been obtained for the individual examples. The variable m is used to denote the total number of examples, which are available in a data set.

$$\begin{aligned}\delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{m} \cdot \sum_j \frac{\sum_i TP_i^j}{\sum_i p_i^j} \text{ , with } \sum_i p_i^j = n \text{ , } \forall j \left(X_j \in C\right) \\ &= \frac{\sum_i \sum_j TP_i^j}{n \cdot m}\end{aligned} \tag{5.48}$$

Proof of Anti-Monotonicity: Equation 5.49 shows, that the equation, which is given in Definition 3.1 is met, when using example-based precision together with the rule-independent evaluation strategy. In accordance with Definition 3.1, it includes two multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$, which share a common body B.

$$\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
\Longrightarrow &\left. \frac{\sum_i \sum_j TP_i^j}{n \cdot m} \right|_{\hat{Y}_s \leftarrow B,T} < \left. \frac{\sum_i \sum_j TP_i^j}{n \cdot m} \right|_{\hat{Y}_p \leftarrow B,T} \leq h_{max} \\
\Longrightarrow &\sum_i \sum_j TP_i^j \Big|_{\hat{Y}_s \leftarrow B,T} < \sum_i \sum_j TP_i^j \Big|_{\hat{Y}_p \leftarrow B,T} \leq TP_{max} \\
\Longrightarrow &\exists i \left(\sum_j TP_i^j \Big|_{\hat{Y}_s \leftarrow B,T} < TP_{max}^i \right) \\
\Longrightarrow &\exists i \left(\sum_j TP_i^j \Big|_{\hat{Y}_a \leftarrow B,T} < TP_{max}^i \right) \text{ , } \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\Longrightarrow &\sum_i \sum_j TP_i^j \Big|_{\hat{Y}_a \leftarrow B,T} < TP_{max} \text{ , } \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\Longrightarrow &\left. \frac{\sum_i \sum_j TP_i^j}{n \cdot m} \right|_{\hat{Y}_a \leftarrow B,T} < h_{max} \text{ , } \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\equiv &\delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h_{max} \text{ , } \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right)
\end{aligned} \tag{5.49}$$

In Equation 5.49, the rule $\hat{Y}_s \leftarrow B$ is assumed to contain additional label attributes beyond those of the rule $\hat{Y}_p \leftarrow B$ in its head. It is further assumed to reach a lower performance than the rule $\hat{Y}_p \leftarrow B$. When rewriting the performance calculations, which are given in the first line of Equation 5.49, with respect to Equation 5.48, it follows from the premise of the proof, that the rule $\hat{Y}_s \leftarrow B$ covers less true positives than the rule $\hat{Y}_p \leftarrow B$. This further implies, that the number of true positives, which are covered by the rule $\hat{Y}_s \leftarrow B$, is less than the maximum number of true positives TP_{max} (cf. Equation 5.49, line 3). When considering each label individually, it can be stated, that for at least one label λ_i the maximum number of true positives TP^i_{max} is not reached by the rule $\hat{Y}_s \leftarrow B$ (cf. Equation 5.49, line 4). When creating rules $\hat{Y}_a \leftarrow B$ by adding additional label attributes to the head $\hat{Y}_s$ of said rule, all of these rules still predict the same value for label λ_i and therefore are not able to reach the maximum number of true positives TP^i_{max} either (cf. Equation 5.49, line 5). As a result, all rules, which result from adding additional label attributes, do not cover as many true positives as the best possible rule is able to cover and therefore none of them reaches the best possible performance h_{max}. As this is in accordance with Definition 3.1, the precision metric is proved to be anti-monotonous, when used together with example-based averaging and the rule-independent evaluation strategy. ■

5.2.1.4 Macro-Averaging

Proof of Anti-Monotonicity: When using macro-averaging for measuring the performance of a rule, heuristic values are obtained for each example and label at first. Afterwards, a single performance is calculated by first example-wise and then label-wise averaging the obtained values, or vice versa. Equation 5.50, which is shown below, illustrates the calculation of a multi-label head rule's precision, when using the first of both averaging orders together with the rule-independent evaluation strategy.

$$\begin{aligned}
\delta_{prec,MM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{n} \cdot \sum_i \left(\frac{1}{m} \cdot \sum_j \frac{TP_i^j}{p_i^j} \right) \text{, with } \frac{TP_i^j}{p_i^j} = TP_i^j \text{ , } \forall i \forall j \\
&= \frac{\sum_i \sum_j TP_i^j}{n \cdot m} \qquad \triangleright \text{ c.f (5.48), last line} \\
&\equiv \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T)
\end{aligned} \tag{5.50}$$

When using macro-averaged precision, the heuristic value, which is obtained for each example and label, is either 0 or 1, depending on whether a true positive is covered, or not. This enables to rewrite Equation 5.50 to be in accordance with Equation 5.48. As Equation 5.48 corresponds to the example-based variant of the precision metric, macro-averaged precision is shown to be equivalent to its example-based counterpart, when using the rule-independent evaluation strategy. Because example-based precision is proved to meet Definition 3.1 in Section 5.2.1.3, macro-averaged precision is implied to meet that definition as well. It therefore can be considered to be anti-monotonous. ■

5.2.2 Recall

In the following subsections, the recall metric, as given in Equation 2.13, is examined in terms of anti-monotonicity, when using micro-averaging, label-based averaging, example-based averaging or macro-averaging together with the rule-independent evaluation strategy. Unlike the precision metric, recall also takes the false negatives, which are covered by a rule, into account. As a result, the performance of a rule does not only depend on the covered examples, but also on the examples, which are not covered by the rule.

5.2.2.1 Micro-Averaging

Equation 5.51 shows, how the micro-averaged recall of a multi-label rule $\hat{Y} \leftarrow B$ is calculated, when using the rule-independent evaluation strategy. As the equation illustrates, the recall metric measures the percentage of true positives among all relevant labels (true positives and false negatives).

$$\delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) = \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} \tag{5.51}$$

In order to show, that micro-averaged recall is anti-monotonous, Lemma 5.2 is utilized. Said lemma states, that the micro-averaged recall of a rule is greater than the performance of another rule, if the first rule covers more true positives – but the same number of false negatives – than the second one. Note, that two rules cover the same number of false positives, if they share a common body and therefore cover the same examples. This is, because the false negatives result from the relevant labels of uncovered instances, regardless of a rule's head.

Lemma 5.2: *If there are two multi-label head rules, which cover the same number of false negatives, but a different number of true positives, the rule, which covers TP_Δ more true positives than the other rule, reaches a higher performance according to the recall metric, when using micro-averaging and a rule-dependent evaluation:*

$$\frac{TP_\Delta + \sum_i \sum_j TP_i^j}{TP_\Delta + \sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} > \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} \tag{5.52}$$

Proof: In order to prove the inequality, which is given in Lemma 5.2, to be hold, it is rewritten in terms of a single fraction at first. As Equation 5.53 shows, rewriting the inequality is based on converting both fractions of the original equation to like quantities by multiplying their denominators with each other and adapting the numerators accordingly.

$$\frac{TP_\Delta + \sum_i \sum_j TP_i^j}{TP_\Delta + \sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} > \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j}$$

$$\frac{TP_\Delta + \sum_i \sum_j TP_i^j}{TP_\Delta + \sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} - \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} > 0 \tag{5.53}$$

$$\frac{TP_\Delta \cdot \sum_i \sum_j FN_i^j}{\left(TP_\Delta + \sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j\right) \cdot \left(\sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j\right)} > 0$$

By proving, that the denominator of the fraction, which is shown in the last line of Equation 5.53, is greater than zero, the inequality, which is part of Equation 5.53, is shown to hold. The denominator is shown to be positive by Equation 5.54 below. It is based on the fact, that the performance of the rule, which covers less true positives than its counterpart, must be less than 1. In this context, the value 1 can

be written as $\frac{TP_\triangle}{TP_\triangle}$. By using cross-multiplication, the inequality, which is given in Equation 5.54, can be converted to be in accordance with the denominator of the fraction given above.

$$\frac{TP_\triangle}{TP_\triangle} > \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} \tag{5.54}$$

$$TP_\triangle \cdot \sum_i \sum_j FN_i^j > 0 \qquad \triangleright \text{ Proofs last line of (5.53) to be hold}$$

■

Proof of Anti-Monotonicity: Equation 5.55 shows, that micro-averaged recall meets the definition of anti-monotonicity, when using the rule-independent evaluation strategy. In accordance with said definition, which is given in Definition 3.1, two multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ take part in the equation. The latter of both rules is assumed to contain additional label attributes, beyond those of the head $\hat{Y}_p$, in its head. Furthermore, it is assumed to have a lower performance than the rule $\hat{Y}_p \leftarrow B$.

$$\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
&\Longrightarrow \left. \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} \right|_{\hat{Y}_s \leftarrow B,T} < \left. \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j TP_i^j + \sum_i \sum_j FN_i^j} \right|_{\hat{Y}_p \leftarrow B,T} \leq h_{max} \\
&\xRightarrow[\text{lemma 5.2}]{\text{w.r.t.}} \sum_i \sum_j TP_i^j \Big|_{\hat{Y}_s \leftarrow B,T} < \sum_i \sum_j TP_i^j \Big|_{\hat{Y}_p \leftarrow B,T} \leq TP_{max} \\
&\Longrightarrow \exists i \left(\sum_j TP_i^j \Big|_{\hat{Y}_s \leftarrow B,T} < TP_{max}^i \right) \\
&\Longrightarrow \exists i \left(\sum_j TP_i^j \Big|_{\hat{Y}_a \leftarrow B,T} < TP_{max}^i \right), \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
&\Longrightarrow \sum_i \sum_j TP_i^j \Big|_{\hat{Y}_a \leftarrow B,T} < TP_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
&\Longrightarrow \left. \frac{\sum_i \sum_j TP_i^j}{\sum_i \sum_j TP_i^j + \sum_i^j FN_i^j} \right|_{\hat{Y}_a \leftarrow B,T} < h_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
&\equiv \delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right)
\end{aligned} \tag{5.55}$$

Both rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$, which take part in Equation 5.55, have an identical body B. Because of this, they cover the same examples. This further implies, that they cover the same number of false negatives. As a result, the difference in their performances must result from the number of true positives each of the rules covers. With respect to Lemma 5.2, it can be stated, that the performance of the rule $\hat{Y}_s \leftarrow B$ is less than the performance of the rule $\hat{Y}_p \leftarrow B$, because it covers less true positives (cf. Equation 5.55, line 3). When considering each label individually, it follows, that there is at least one label λ_i, for which the rule $\hat{Y}_s \leftarrow B$ does not cover the maximum number of true positives TP^i_{max} (cf. Equation 5.55, line 4). When adding additional label attributes to the rule's head, the prediction for that particular label remains the same. Therefore no rule $\hat{Y}_a \leftarrow B$, which results from adding additional label attributes to the head of the rule $\hat{Y}_s \leftarrow B$, is able to reach the maximum number of true positives TP_{max} or the maximum performance h_{max} (cf. Equation 5.55, line 6 and 7). This corresponds to the anti-monotonicity property as specified in Definition 3.1. Equation 5.55 therefore shows, that Definition 3.1 is met, when using the recall metric, together with micro-averaging and the rule-independent evaluation strategy, for measuring the performances of multi-label head rules. As a result, said evaluation strategy is proved to meet the properties of anti-monotonicity. ■

5.2.2.2 Label-based Averaging

When using label-based averaging, the performance of a rule is calculated by first obtaining a heuristic value per label and averaging the obtained values afterwards. Equation 5.56 shows, how the recall of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated, when using said averaging strategy together with a rule-independent evaluation.

$$\delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) = \frac{1}{n} \cdot \sum_i \frac{\sum_j TP_i^j}{\sum_j P_i^j} \tag{5.56}$$

Proof of Anti-Monotonicity: The proof, which is given in Equation 5.57, shows, that the recall metric is anti-monotonous, when used together with label-based averaging and the rule-independent evaluation strategy. It is based on the consideration, that the prediction for at least one label must be non-optimal, if the overall performance of a rule is less than the best possible performance h_{max}. According to the definition of anti-monotonicity, which is given in Definition 3.1, Equation 5.57 includes two multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ of which the first one is assumed to outperform the latter one. This implies, that the rule $\hat{Y}_s \leftarrow B$ – whose head contains additional label attributes beyond those of the head $\hat{Y}_p$ – cannot reach the best possible performance h_{max} (cf. Equation 5.57, line 2). Consequently, there is a label λ_i, for which the rule $\hat{Y}_s \leftarrow B$ does not reach the performance h^i_{max}. When adding additional label attributes to the rule's head, the prediction for the label λ_i remains unchanged. This implies, that the performance, which is reached by a rule $\hat{Y}_a \leftarrow B$, which results from adding additional label attributes to the head $\hat{Y}_s$, is still less than h^i_{max} and therefore not optimal (cf. Equation 5.57, line 4). Because of this, it is shown, that by adding additional label attributes to the head of rule $\hat{Y}_s \leftarrow B$, the best possible performance h_{max} cannot be reached. As this is in accordance with the equation, which is given as part of Definition 3.1, Equation 5.57 proves the properties of anti-monotonicity to be met in case of using label-based recall together with the rule-dependent evaluation strategy.

$$
\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta(\hat{Y}_s \leftarrow B, T) < \delta(\hat{Y}_p \leftarrow B, T) \\
\Longrightarrow &\frac{1}{n} \cdot \sum_i \left. \frac{\sum_j TP_i^j}{\sum_j P_i^j} \right|_{\hat{Y}_s \leftarrow B,T} < \frac{1}{n} \cdot \sum_i \left. \frac{\sum_j TP_i^j}{\sum_j P_i^j} \right|_{\hat{Y}_p \leftarrow B,T} \leq h_{max} \\
\Longrightarrow &\exists i \left(\left. \frac{\sum_j TP_i^j}{\sum_j P_i^j} \right|_{\hat{Y}_s \leftarrow B,T} < h_{max}^i \right) \\
\Longrightarrow &\exists i \left(\left. \frac{\sum_j TP_i^j}{\sum_j P_i^j} \right|_{\hat{Y}_a \leftarrow B,T} < h_{max}^i \right), \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
\Longrightarrow &\frac{1}{n} \cdot \sum_i \left. \frac{\sum_j TP_i^j}{\sum_j P_i^j} \right|_{\hat{Y}_a \leftarrow B,T} < h_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right) \\
\equiv &\delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a \right)
\end{aligned}
\tag{5.57}
$$

■

5.2.2.3 Example-based Averaging

According to the definition of example-based averaging, which is given in Equation 2.9, the performance of a rule is calculated by first obtaining a heuristic value for each example and averaging the results afterwards. The example, which is given in the following, reveals, that the properties of anti-monotonicity are not met, when using said averaging strategy for measuring the rule-independent recall of multi-label head rules.

Disproof of Anti-Monotonicity: The counterexample, which is given in the following, is based on the exemplary label vectors, which are shown in Table 6. The label space, which is used in the example, includes four labels $\mathbb{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. Similar to previous examples, some of the examples in Table 6 are assumed to be covered by a rule's body and some are not.

		λ_1	λ_2	λ_3	λ_4
Not covered	Y_1	0	1	1	1
	Y_2	1	1	1	1
	Y_3	0	1	0	0
Covered	Y_4	1	0	0	0
	Y_5	1	0	0	0
	Y_6	0	0	1	1

Table 6: Exemplary label vectors of training examples used by Figure 4 and given the label space $\mathbb{L} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. Some examples are assumed to be covered by a given body, some are not.

In Figure 4, a search tree, which corresponds to an exhaustive search through the label space $\mathbb{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ is shown. The search is based on the training examples, which are shown in Table 6. For evaluating potential multi-label heads, example-based recall is used together with the rule-independent evaluation strategy. Whenever adding a label attribute to a head causes the performance of the resulting rule to decrease, the corresponding edge in Figure 4 is highlighted by using a red arrow (→). If the performance increases or remains constant instead, green (→) and black arrows (→) are used accordingly.

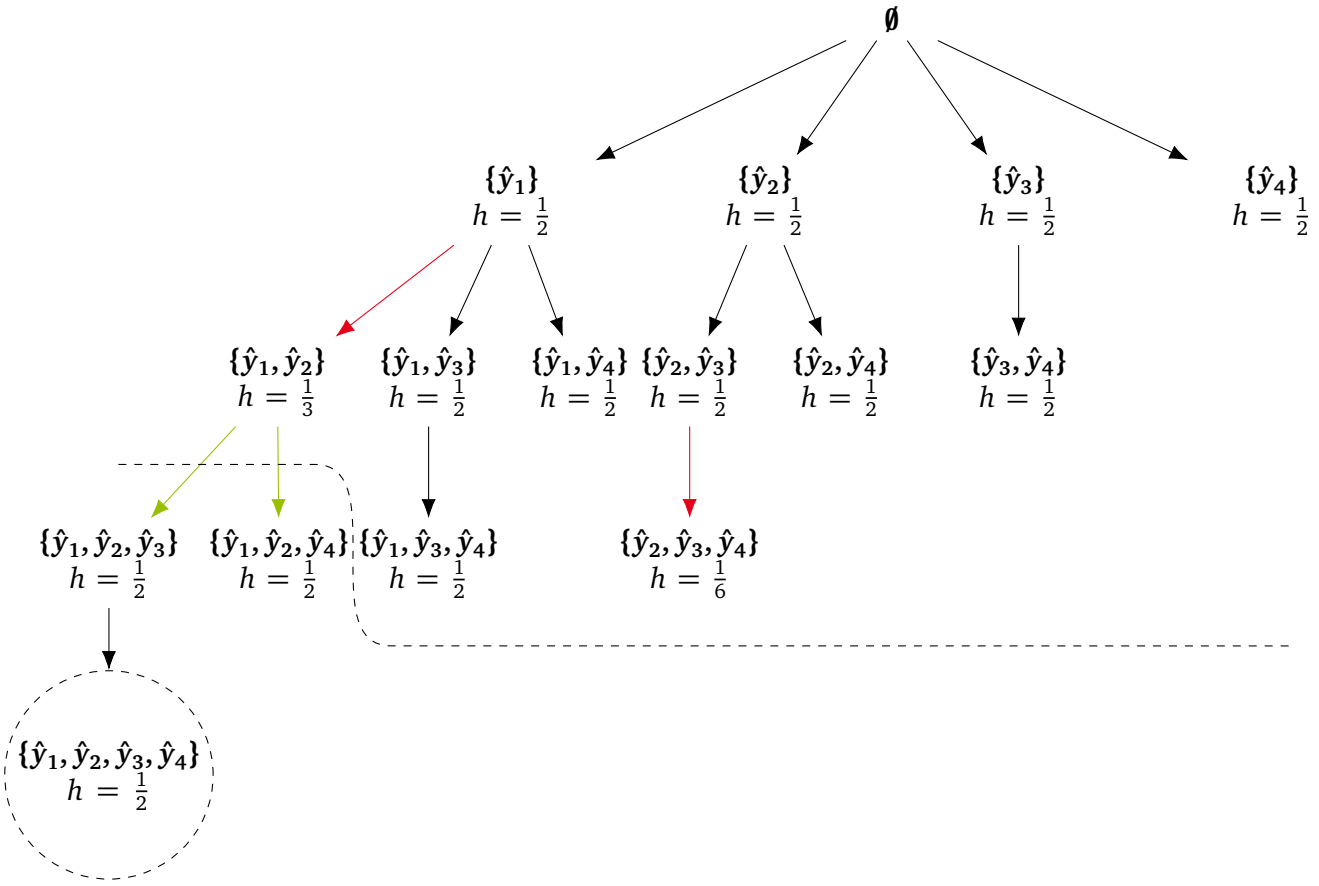

Figure 4: Search through the label space for finding the best multi-label rule head given the examples in Table 6 and using label-based recall, together with the rule-independent evaluation strategy, for performance evaluation. The dashed line (- - -) indicates the label combinations, which are left out, when pruning the search according to the properties of anti-monotonicity as given in Definition 3.1.

According to Figure 4, the label combination $\{\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4\}$, which reaches a performance of $\frac{1}{2}$, is considered to be the best solution. However, when pruning the search according to the anti-monotonicity property – as indicated by the dashed line (- - -) –, this particular label combination is not found. This is, because the performance decreases from $\frac{1}{2}$ to $\frac{1}{3}$, when adding the label attribute $\hat{y}_2$ to the head $\{\hat{y}_1\}$. According to the anti-monotonicity property, the heads, which result from adding additional label attributes – namely the head$\{\hat{y}_1, \hat{y}_2, \hat{y}_3\}$, as well as $\{\hat{y}_1, \hat{y}_2, \hat{y}_4\}$ and $\{\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4\}$ –, cannot reach the best possible performance. Therefore these label combinations are not considered by a pruned search. However, as Figure 4 shows, adding the label attribute $\hat{y}_3$ to the head $\{\hat{y}_1, \hat{y}_2\}$ causes the best possible performance of $\frac{1}{2}$ to be reach, which contradicts the definition of anti-monotonicity. Because of this, the recall metric is disproved to be anti-monotonous according to Definition 3.1, when used together with example-based averaging and when utilizing the rule-independent evaluation strategy. ■

5.2.2.4 Macro-Averaging

According to Equation 2.11, the macro-averaged performance of a rule is calculated by obtaining a heuristic value for each example and label at first. By first calculating the example-wise arithmetic mean and then the label wise arithmetic mean, or vice versa, the individual heuristic values are finally averaged in order to obtain a single performance. The proof, which is given in the following, uses the first of both averaging orders in order to show, that macro-averaged recall is equivalent to macro-averaged precision, when using a rule-independent evaluation.

Proof of Anti-Monotonicity: In Equation 5.58, the calculation of a multi-label head rule's recall, when using macro-averaging and the rule-independent evaluation strategy, is illustrated.

$$\begin{aligned}
\delta_{rec,MM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) =& \frac{1}{n} \cdot \sum_i \left(\frac{1}{m} \cdot \sum_j \frac{TP_i^j}{P_i^j} \right) \text{, with } \frac{TP_i^j}{P_i^j} = TP_i^j \text{ , } \forall i \forall j \\
=& \sum_i \sum_j \frac{TP_i^j}{n \cdot m} && \triangleright \text{ c.f (5.50), line 2} \\
\equiv& \delta_{prec,MM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) \\
\equiv& \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T)
\end{aligned} \tag{5.58}$$

As illustrated by Equation 5.58, the heuristic value, which is calculated for an individual example and label, is either 0 or 1, depending on whether a true positive is covered, or not. Because of this, the equation can be converted into a single fraction, which is in accordance with the second line of Equation 5.50. Said equation corresponds to the calculation of a rule's performance according to macro-averaged precision. Consequently, it follows, that using macro-averaged recall for measuring the performance of a rule is equivalent to using macro-averaged precision, if the rule-independent evaluation strategy is used. Because, macro-averaged precision is proved to be anti-monotonous in Section 5.2.1.4, when using a rule-independent evaluation, the macro-averaged variant of the recall metric is implied to meet Definition 3.1 as well. Furthermore, it can also be considered to be equivalent to example-based precision, because the latter is shown to be equivalent to the macro-averaged variant of the precision metric in Section 5.2.1.4 as well. ■

5.2.3 Hamming Accuracy

In this section, the accuracy metric is examined in terms of anti-monotonicity, when using the rule-independent evaluation strategy. The consideration, which are given in this section, are similar to the ones, which are given in Section 5.2.1 with respect to the precision metric. As the proofs, which are given in the following subsections, reveal, hamming accuracy meets the definition of anti-monotonicity regardless of whether micro-averaging, label-based averaging, example-based averaging or macro-averaging is used.

5.2.3.1 Micro-Averaging

According to Equation 2.14, the hamming accuracy metric measures the percentage of correctly predicted relevant and irrelevant labels among all labels. An example of using that evaluation strategy for searches through the label space is given in Figure 1 of Chapter 3. Equation 5.59, which is shown below, illustrates, how the performance of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated when using said evaluation function together with micro-averaging and the rule-independent evaluation strategy. The equation

can be converted to only depend on true positives and true negatives, because the number of all labels equals the number of examples m, which are contained by the respective data set, times the number of available labels n. Furthermore, the number of true negatives, which are covered by two rules, is equal, if both rules share an identical body. This is, because the true negatives depend on examples, which are not covered by a rule's body, whereas the true positives result from a rules' predictions on covered examples.

$$\begin{aligned}\delta_{hamm,mm,\amalg}(\hat{Y} \leftarrow B, T) &= \frac{\sum_i \sum_j \left(TP_i^j + TN_i^j\right)}{\sum_i \sum_j \left(P_i^j + N_i^j\right)} \text{ , with } \sum_i \sum_j \left(P_i^j + N_i^j\right) = n \cdot m \\ &= \frac{\sum_i \sum_j \left(TP_i^j + TN_i^j\right)}{n \cdot m}\end{aligned} \tag{5.59}$$

Proof of Anti-Monotonicity: Equation 5.60 shows, that anti-monotonicity, as specified in Definition 3.1, is met by micro-averaged hamming accuracy, when used together with the rule-independent evaluation strategy. According to the equation, which is given in Definition 3.1, in Equation 5.60 two multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ take part.

$$\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta_{hamm,mm,\amalg}(\hat{Y}_s \leftarrow B, T) < \delta_{hamm,mm,\amalg}(\hat{Y}_p \leftarrow B, T) \\
\Longrightarrow &\left.\frac{\sum_i \sum_i \left(TP_i^j + TN_i^j\right)}{n \cdot m}\right|_{\hat{Y}_s \leftarrow B,T} < \left.\frac{\sum_i \sum_j \left(TP_i^j + TN_i^j\right)}{n \cdot m}\right|_{\hat{Y}_p \leftarrow B,T} \leq h_{max} \\
\Longrightarrow &\sum_i \sum_j \left.TP_i^j\right|_{\hat{Y}_s \leftarrow B,T} < \sum_i \sum_j \left.TP_i^j\right|_{\hat{Y}_p \leftarrow B,T} \leq TP_{max} \\
\Longrightarrow &\exists i \left(\sum_j \left.TP_i^j\right|_{\hat{Y}_s \leftarrow B,T} < TP_{max}^i \right) \\
\Longrightarrow &\exists i \left(\sum_j \left.TP_i^j\right|_{\hat{Y}_a \leftarrow B,T} < TP_{max}^i \right), \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\Longrightarrow &\sum_i \sum_j \left.TP_i^j\right|_{\hat{Y}_a \leftarrow B,T} < TP_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\Longrightarrow &\left.\frac{\sum_i \sum_j \left(TP_i^j + TN_i^j\right)}{n \cdot m}\right|_{\hat{Y}_a \leftarrow B,T} < h_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\equiv &\delta_{hamm,mm,\amalg}(\hat{Y}_a \leftarrow B, T) < h_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right)
\end{aligned} \tag{5.60}$$

Whereas both rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$, which take part in Equation 5.60, share the same body B, the head $\hat{Y}_s$ is assumed to contain additional label attributes besides those, the head $\hat{Y}_p$ consists of. Furthermore, the rule $\hat{Y}_p \leftarrow B$ is assumed to outperform the rule $\hat{Y}_p \leftarrow B$, which implies that the latter cannot reach the best possible performance h_{max} (cf. Equation 5.60, line 1 and 2). When writing the calculation of both rules' performances according to Equation 5.59, the performances are denotated as fractions with identical denominators. Because of this, the difference in both rules' performances can only result from the true positives and true negatives, the respective numerators are calculated from. However, as two rules with identical bodies cover the same number of true negatives, it follows, that the reason for the performance of the rule $\hat{Y}_s \leftarrow B$ to be lower than that of the rule $\hat{Y} \leftarrow B$ must be, that it covers less true positives (cf. Equation 5.60, line 3). When considering each available label individually, this implies, that for at least one label λ_i the maximum number of true positives TP^i_{max} is not reached by the rule $\hat{Y}_s \leftarrow B$ (cf. Equation 5.60, line 4). Even when adding additional label attributes to the head of said rule, the prediction for the label λ_i remains unchanged and therefore the maximum number of true positives TP_{max} cannot be reached by such rule $\hat{Y}_a \leftarrow B$ either (cf. Equation 5.60, line 5 and 6). From that observation follows, that no rule $\hat{Y}_a \leftarrow B$, which results from adding additional label attributes to the head of the rule $\hat{Y}_s \leftarrow B$, is able to reach the best possible performance h_{max} (cf. Equation 5.60, line 7 and 8). This corresponds to anti-monotonicity property as given in Definition 3.1. Consequently, Equation 5.60 proves the definition of anti-monotonicity to be met, if micro-averaged hamming accuracy is used for measuring the performance of multi-label head rules using the rule-independent evaluation strategy. ■

5.2.3.2 Label-based Averaging

When using label-based averaging according to Equation 2.10 for measuring the performance of a multi-label head rule, a heuristic value is obtained per available label at first. The overall performance of the rule finally calculates as the arithmetic mean of all obtained values. The following proof shows, that using said averaging strategy for measuring the rule-independent performance of a rule, according the hamming accuracy metric, is equivalent to using micro-averaging.

Proof of Anti-Monotonicity: In Equation 5.61, the calculation of a multi-label head rule's performance, using the label-based hamming accuracy metric together with the rule-independent evaluation strategy, is illustrated.

$$
\begin{aligned}
\delta_{hamm,mM,\amalg}(\hat{Y} \leftarrow B, T) &= \frac{1}{n} \cdot \sum_i \frac{\sum_j \left(TP_i^j + TN_i^j\right)}{\sum_j \left(P_i^j + N_i^j\right)} \text{ , with } \sum_j \left(P_i^j + N_i^j\right) = m \text{ , } \forall i \\
&= \frac{1}{n} \cdot \sum_i \frac{\sum_j \left(TP_i^j + TN_i^j\right)}{m} \\
&= \frac{\sum_i \sum_j \left(TP_i^j + TN_i^j\right)}{n \cdot m} \qquad \triangleright \text{ c.f (5.59), last line} \\
&\equiv \delta_{hamm,mm,\amalg}(\hat{Y} \leftarrow B, T)
\end{aligned}
\tag{5.61}
$$

According to Equation 5.61, the heuristic value, which is calculated for an individual label, measures the percentage of examples for which the respective label is predicted correctly. It can be written as a fraction, whose numerator consists of the covered true positives and true negatives, whereas the denominator represents the number of examples m. By further rewriting the equation, it can be converted to be in accordance with the last line of Equation 5.59, which corresponds to using the micro-averaged variant of the hamming accuracy metric. As a result, label-based and micro-averaged hamming accuracy are shown to be equivalent, when used in terms of the rule-independent evaluation strategy. As the latter is shown to meet the properties of anti-monotonicity, the label-based variant is implied to be anti-monotonous according to Definition 3.1 as well. ■

5.2.3.3 Example-based Averaging

According to the definition of example-based averaging, which is given in Equation 2.9, when using said averaging strategy, a heuristic value is calculated for each example at first. Afterwards, by averaging the obtained values, the overall performance of a rule is computed. Such as the prove, which is given in the previous section, the proof, which is given in the following, shows, that using example-based averaging for measuring the rule-independent hamming accuracy of a rule is equivalent to using micro-averaging.

Proof of Anti-Monotonicity: Equation 5.62 shows, how the performance of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated according to the hamming accuracy metric, when using example-based averaging together with the rule-independent evaluation strategy. According to Equation 5.62, the heuristic value, which is calculated for an individual example, measures the percentage of labels, which are predicted correctly by the rule. This corresponds to a fraction, whose numerator consists of the covered true positives and true negatives, whereas the denominator equals the number of available labels n (cf. Equation 5.62, line 2). By rewriting the equation, it can be converted to the same form as used in the last line of Equation 5.59. This proves, that the example-based hamming accuracy of a rule is equal to the micro-averaged hamming accuracy, when using the rule-independent evaluation strategy.

$$
\begin{aligned}
\delta_{hamm,Mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{m} \cdot \sum_j \frac{\sum_i \left(TP_i^j + TN_i^j\right)}{\sum_i \left(P_i^j + N_i^j\right)} \text{ , with } \sum_i \left(P_i^j + N_i^j\right) = n \text{ , } \forall j \\
&= \frac{1}{m} \cdot \sum_j \frac{\left(TP_i^j + TN_i^j\right)}{n} \\
&= \frac{\sum_i \sum_j \left(TP_i^j + TN_i^j\right)}{n \cdot m} \qquad \triangleright \text{ c.f (5.59), last line} \\
&\equiv \delta_{hamm,mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{hamm,mM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T)
\end{aligned}
\tag{5.62}
$$

From Equation 5.62 follows, that the example-based variant of the hamming accuracy metric is equivalent to its micro-averaged counterpart, if the rule-dependent evaluation strategy is used. As the latter of both evaluation functions is shown to be anti-monotonous in Section 5.2.3.1, when using example-based averaging, it is implied, that the definition of anti-monotonicity, which is given in Definition 3.1, is met as well. Furthermore, as it is shown in Section 5.2.3.2, that using label-based averaging for measuring the rule-independent hamming accuracy of a rule is equivalent to using micro-averaging, example-based averaging is also equivalent to that variant. ■

5.2.3.4 Macro-Averaging

According to Equation 2.11, when using macro-averaging for measuring the performance of a rule, a heuristic value is obtained per example and label at first. In order to calculate the overall performance of the rule, the obtained values are first example-wise averaged and then label-wise averaged, or vice versa. The first of both orders is used by the following proof to show the equivalence of macro- and micro-averaging, when used together with the hamming accuracy metric and the rule-independent evaluation strategy.

Proof of Anti-Monotonicity: Equation 5.63 shows, how the macro-averaged hamming accuracy of a multi-label head rule $\hat{Y} \leftarrow B$ calculates, when using a rule-independent evaluation. The heuristic value, which is obtained for each example and label either evaluates to 1, if a true positive or true negative is covered, or to 0 otherwise. This allows to eliminate the denominator of the fraction, which is shown in the first line of Equation 5.63. By further converting the equation, it can be rewritten to match the term, which is shown in the last line of Equation 5.59. This proves micro- and macro-averaging to be equivalent, if used for measuring the rule-independent hamming accuracy of a multi-label head rule.

$$
\begin{aligned}
\delta_{hamm,MM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{n} \cdot \sum_i \left(\frac{1}{m} \cdot \sum_j \frac{TP_i^j + TN_i^j}{P_i^j + N_i^j} \right) \text{, with } P_i^j + N_i^j = 1 \text{ , } \forall i \forall j \\
&= \frac{1}{n} \cdot \sum_i \left(\frac{1}{m} \cdot \sum_j TP_i^j + TN_i^j \right) \\
&= \frac{1}{n} \cdot \sum_i \frac{\sum_j \left(TP_i^j + TN_i^j \right)}{m} \\
&= \frac{\sum_i \sum_j \left(TP_i^j + TN_i^j \right)}{n \cdot m} \qquad \triangleright \text{ c.f (5.59), last line} \\
&\equiv \delta_{hamm,mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{hamm,mM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{hamm,Mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T)
\end{aligned}
\tag{5.63}
$$

As a result of Equation 5.63, which shows micro- and macro-averaging to be equivalent, when being used for measuring the rule-independent hamming accuracy of multi-label head rules, the latter of both variants is implied to be anti-monotonous. This is, because micro-averaged hamming accuracy is shown to meet Definition 3.1 in Section 5.2.3.1. Furthermore, using the label-based or example-based averaging hamming accuracy metric together with the rule-independent evaluation strategy is shown to be equivalent to using the micro-averaging variant in Section 5.2.3.2, respectively Section 5.2.3.3. Consequently, the macro-averaged variant of the hamming accuracy is equivalent to using these averaging strategies as well. ■

5.2.4 F-Measure

In the individual subsections of this section, the F-Measure is examined in terms of anti-monotonicity, depending on whether micro-averaging, label-based averaging, example-based averaging or macro-averaging is used. As already mentioned in Section 5.1.4, the F-Measure is defined as the weighted harmonic mean of precision and recall. As the F-Measure is equivalent to the precision metric, if its β-parameter is set to 0, only cases where $\beta > 0$ are considered in the present section. In the proofs, which are given throughout the following subsections, the F-Measure is often written in terms of the harmonic mean operation H as defined in Equation 5.28. Such as all Pythagorean means, the harmonic mean operation meets the averaging property (cf. Equation 5.29), as well as the value preservation property (cf. Equation 5.30).

5.2.4.1 Micro-Averaging

The proof, which is given below, shows, that the micro-averaged variant of the F-Measure meets the anti-monotonicity property, when used together with the rule-independent evaluation strategy. It is based on the fact, that micro-averaged recall and precision are anti-monotonous as well, when used for a rule-independent evaluation, as it is proved in Section 5.2.2, respectively Section 5.2.1. Furthermore it uses the notation, which is given in Equation 5.31, for denoting the best possible performance according to a certain evaluation function.

Proof of Anti-Monotonicity: As the premise of the proof at hand, the best possible performance according to the recall metric is assumed to be greater or equal than the best possible performance according to the precision metric. Equation 5.64, which is given below, illustrates this assumption. It can be made without loss of generality, because the proof, which is given in the following, can easily be adopted to an alternative premise: If the best performance according to the precision metric is assumed to be greater or equal than the best possible performance according to the recall metric, the corresponding proof would be similar.

$$h_{max}^{rec} \geq h_{max}^{prec} \tag{5.64}$$

Furthermore, as already pointed out by Equation 5.32, the best possible performance according to the F-Measure cannot be greater than the maximum of the best possible performances according to recall and precision. Consequently, the inequality, which is given in Equation 5.65 below, holds.

$$h_{max}^{F} \leq max\left(h_{max}^{rec}, h_{max}^{prec}\right) \tag{5.65}$$

Equation 5.66 proves the equation, which is given in Definition 3.1, to be met in case of micro-averaged F-Measure, when using the rule-independent evaluation strategy. It is based on rewriting the F-Measure of two multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ in terms of the harmonic mean operation H (cf. Equation 5.66, line 2). In accordance with the equation, which is given in Definition 3.1, both multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ share a common body B. However, the first of both rules is assumed to outperform the second one due to their dissimilar heads: The head $\hat{Y}_s$ is assumed to contain additional label attributes beyond those of the head $\hat{Y}_p$ and therefore the subset relationship $\hat{Y}_p \subset \hat{Y}_s$ holds.

$$
\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta_{F,mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{F,mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
\equiv &\hat{Y}_p \subset \hat{Y}_s \wedge H\left(\delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T), \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T)\right) \\
< &H\left(\delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T), \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T)\right) \\
\xRightarrow[\text{and (5.30)}]{\text{w.r.t. (5.30)}} &\delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
&\vee \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
\xRightarrow[\delta rec,mm,\perp\!\!\!\perp \text{ and } \delta_{prec,mm,\perp\!\!\!\perp}]{\text{w.r.t. anti-monotonicity of}} &\left(\delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h^{rec}_{max} \wedge \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B) < h^{rec}_{max}\right) \\
&\vee \left(\delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h^{rec}_{max} \wedge \delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B) \leq h^{rec}_{max}\right) , \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\xRightarrow[\text{and (5.30)}]{\text{w.r.t. (5.30)}} &H\left(\delta_{rec,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T), \delta_{prec,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T)\right) < h^{F}_{max} \leq h^{rec}_{max} , \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\equiv &\delta_{F,mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h^{F}_{max} , \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right)
\end{aligned}
\tag{5.66}
$$

As already mentioned, the F-Measure calculates as the harmonic mean of precision and recall. As a result, the averaging and value preservation properties (cf. Equation 5.29 and Equation 5.30) are given. Because of this, either the recall or the precision of the rule $\hat{Y}_s \leftarrow B$ must be lower than the corresponding heuristic value of the rule $\hat{Y}_p \leftarrow B$ (cf. Equation 5.66, line 4 and 5). In Section 5.2.2 and Section 5.2.1 it is shown that both, the recall metric, as well as the precision metric, fulfill the properties of anti-monotonicity, when using the rule-independent evaluation strategy. Consequently, if the recall or precision of the rule $\hat{Y}_s \leftarrow B$ is less than that of the rule $\hat{Y}_p \leftarrow B$, the recall, respectively precision, of any multi-label head rule $\hat{Y}_a \leftarrow B$, which result from adding additional label attribute to the head $\hat{Y}_s$, cannot reach the best possible performance h^F_{max}. Furthermore, due to the premise of the proof, the best possible performance h^{rec}_{max} can be considered as an upper border for the performance of the rule $\hat{Y}_a \leftarrow B$, regardless of it is measured by using the recall or precision metric (cf. Equation 5.66, line 6 and 7). From the averaging and value preservation properties of the harmonic mean operation follows, that the F-Measure of the rule $\hat{Y}_a \leftarrow B$ cannot reach the performance h^{rec}_{max}. As the performance h^F_{max} cannot be greater than h^{rec}_{max} due to Equation 5.65, the F-Measure of said rule is further guaranteed to be less than h^F_{max} (cf. Equation 5.66, line 8 and 9). According to the given argumentation, the definition of anti-monotonicity is fulfilled, because if the performance of a rule decreases after adding an additional label attribute to its head, by adding even more label attributes, the best performance h^F_{max} cannot be reached anymore. The F-Measure is therefore shown to be anti-monotonous, according to Definition 3.1, if it is used together with micro-averaging and the rule-independent evaluation strategy and if the β-parameter is set to a value greater than 0. If $\beta = 0$ instead, the micro-averaged F-Measure is equivalent to the micro-averaged variant of the precision metric, which is proved to be anti-monotonous in Section 5.2.1.1. ■

5.2.4.2 Label-based Averaging

Proof of Anti-Monotonicity: In Equation 5.67, a prove, which shows, that the label-based F-Measure meets the definition of anti-monotonicity, when used for measuring the rule-independent performance of a multi-label head rule, is given. It is similar to the proof, which is shown in Equation 5.66 of the previous section. Such as that proof, Equation 5.67 is based on the premise, which is given in Equation 5.64. It states without loss of generality, that the best performance according to the recall metric is greater or equal than the best performance according to the precision metric.

$$
\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta_{F,mM,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{F,mM,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
\equiv &\hat{Y}_p \subset \hat{Y}_s \wedge H\left(\delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T), \delta_{prec,mM,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T)\right) \\
< &H\left(\delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T), \delta_{prec,mM,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T)\right) \\
\xRightarrow[\text{and (5.30)}]{\text{w.r.t. (5.30)}} &\delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
&\vee \delta_{prec,mM,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{prec,mM,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
\xRightarrow[\delta rec,mM,\perp\!\!\!\perp \text{ and } \delta_{prec,mM,\perp\!\!\!\perp}]{\text{w.r.t. anti-monotonicity of}} &\left(\delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h^{rec}_{max} \wedge \delta_{prec,mM,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h^{rec}_{mac}\right) \\
&\vee \left(\delta_{prec,mM,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h^{rec}_{max} \wedge \delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B) \leq h^{rec}_{max}\right), \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\xRightarrow[\text{and (5.30)}]{\text{w.r.t. (5.30)}} &H\left(\delta_{rec,mM,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T), \delta_{prec,mM,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T)\right) < h^{F}_{max} \leq h^{rec}_{mac}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
\equiv &\delta_{F,mM,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h^{F}_{max}, \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right)
\end{aligned}
\tag{5.67}
$$

Such as Equation 5.66, Equation 5.67 is based on rewriting the F-Measure in terms of the harmonic mean of recall and precision. As both, label-based recall, as well as label-based precision, are shown to be anti-monotonous in Section 5.2.2, respectively Section 5.2.1, it follows, that the F-Measure meets Definition 3.1 as well. As a result, if $\beta > 0$, the micro-averaged F-Measure can considered to be anti-monotonous, when using the rule-independent evaluation strategy. When setting $\beta = 0$, the label-based F-Measure is equivalent to the label-based precision metric. A proof, which shows, that the latter evaluation strategy is anti-monotonous, when using the rule-independent evaluation strategy, can be found in Section 5.2.1.2. ■

5.2.4.3 Example-based Averaging

In the following, it is proved, that example-based F-Measure is anti-monotonous when using the rule-independent evaluation strategy. The proof is based on rewriting the calculation of a multi-label head rule's performance according to said evaluation strategy in terms of the weighted harmonic mean H as shown in Equation 5.68. As the equation reveals, the recall and precision of an individual example both evaluate to 0, if no true positives are covered for that particular example. If at least on true positive is covered, the recall evaluates to 1 instead. As a consequence, the recall is constant for all examples for

which true positives are covered, resulting in the F-Measure for these examples to be greater than the precision, due to the averaging property of the harmonic mean operation (cf. Equation 5.29).

$$\delta_{F,Mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) = \frac{1}{m} \cdot \sum_j \begin{cases} H\left(1, \frac{\sum_i TP_i^j}{\sum_i p_i^j}\right), & \text{if } \sum_i TP_i^j > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.68}$$

Proof of Anti-Monotonicity: Equation 5.69 proves, that the F-Measure meets the definition of anti-monotonicity, when used together with example-based averaging and a rule-independent evaluation. According to the equation, which is given in Definition 3.1, two multi-label head rules $\hat{Y}_p \leftarrow B$ and $\hat{Y}_s \leftarrow B$ take part in Equation 5.69. It is assumed, that the first of both rules outperforms the second one and that the head $\hat{Y}_s$ contains other label attributes in addition to those of the head $\hat{Y}_p$. Furthermore, the proof is based on the premise, that the F-Measure of the rule $\hat{Y}_p \leftarrow B$ is greater than the F-Measure of the rule $\hat{Y}_s \leftarrow B$. This implies, that the precision of the first rule must be greater than that of the latter one as well(cf. Equation 5.69, line 2). This implication is based on Equation 5.68, which states, that the harmonic mean of the recall and precision, which are obtained for individual examples, solely depends on the precision: If the precision is 0, the recall is 0 as well, resulting in the harmonic mean to evaluate to 0. If the precision is greater than 0 instead, the recall is always 1, resulting in the harmonic mean to solely depend on the measured precision. Because in Section 5.2.1.3 the example-based precision metric is shown to be anti-monotonous, when used together with the rule-independent evaluation strategy, the performance of any multi-label head rule $\hat{Y}_a \leftarrow B$, which results from adding additional label attributes to the head of the rule $\hat{Y}_s \leftarrow B$, must be less than the best possible performance h_{max} (cf. Equation 5.69, line 3). Consequently, due to its averaging property, the harmonic mean of precision and recall must be less than h_{max} as well (cf. Equation 5.69, line 4).

$$\begin{aligned}
&\hat{Y}_p \subset \hat{Y}_s \wedge \delta_{F,Mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{F,Mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
&\xRightarrow[\text{and (5.69)}]{\text{w.r.t. (5.30), (5.29)}} \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y}_s \leftarrow B, T) < \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y}_p \leftarrow B, T) \\
&\xRightarrow[\text{of } \delta_{prec,Mm,\perp\!\!\!\perp}]{\text{w.r.t. anti-monotonicity}} \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h_{max}\ ,\ \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
&\xRightarrow[\text{and (5.29)}]{\text{w.r.t. (5.30)}} H\left(\delta_{rec,Mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B), \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B)\right) < h_{max}\ ,\ \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right) \\
&\equiv \delta_{F,Mm,\perp\!\!\!\perp}(\hat{Y}_a \leftarrow B, T) < h_{max}\ ,\ \forall \hat{Y}_a \left(\hat{Y}_s \subset \hat{Y}_a\right)
\end{aligned} \tag{5.69}$$

Equation 5.69 proves, that the equation, which is given in Definition 3.1, holds in case of using the example-based F-Measure for measuring the rule-independent performance of multi-label head rules. Consequently, if the β-parameter is set to a value greater than 0, said evaluation strategy can be considered to be anti-monotonous. If the β-parameter is set to 0, the F-Measure is equivalent to precision. The example-based variant of the precision metric is shown to be anti-monotonous, when using the rule-independent evaluation strategy, in Section 5.2.1.3. ■

5.2.4.4 Macro-Averaging

According to the definition of macro-averaging, which is given in Equation 2.11, when using said averaging strategy, a heuristic value is calculated for each example and label at first. Afterwards, in order to compute the overall performance of a multi-label head rule, the obtained values are averaged example- and label-wise, or vice versa. The first of both averaging orders is used in the following proof. It shows, that the macro-averaged F-Measure is anti-monotonous, when used together the with rule-independent evaluation strategy.

Proof of Anti-Monotonicity: Equation 5.70 shows, how the rule-independent performance of a multi-label head rule $\hat{Y} \leftarrow B$ is calculated according to the F-Measure, when using macro-averaging. The heuristic value, which is obtained for each example and label corresponds to the (weighted) harmonic mean of recall and precision. As already mentioned in Section 5.2.2.4 and Section 5.2.1.4, both, recall and precision, either evaluate to 0 or 1 for an individual example and label, depending on whether a true positive is covered, or not (cf. Equation 5.70, line 2). Because of this, for each example and label, recall and precision are equal and therefore – due to the value preservation property of the harmonic mean operation (cf. Equation 5.30) – the F-Measure evaluates to the same heuristic value as well. As a result, Equation 5.70 can be rewritten as the fraction of true positives among all labels, which are contained in the data set (cf. Equation 5.70, line 3). As the rewritten equation is in accordance with the second line of Equation 5.58, as well as with the second line of Equation 5.50, the macro-averaged F-Measure is implied to be equivalent to macro-averaged recall and precision, when using the rule-independent evaluation strategy.

$$
\begin{aligned}
\delta_{F,MM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) &= \frac{1}{n} \cdot \sum_i \left(\frac{1}{m} \cdot \sum_j H\left(\frac{TP_i^j}{P_i^j}, \frac{TP_i^j}{p_i^j} \right) \right) \text{, with } \frac{TP_i^j}{P_i^j} = \frac{TP_i^j}{p_i^j} = TP_i^j \text{ , } \forall i \forall j \\
&= \frac{1}{n} \cdot \sum_i \left(\frac{1}{m} \cdot \sum_j H\left(TP_i^j, TP_i^j \right) \right) && \triangleright \text{ (5.30) applies} \\
&= \frac{\sum_i \sum_j TP_i^j}{n \cdot m} && \triangleright \text{ c.f (5.50), line 2 and (5.58), line 2} \\
&\equiv \delta_{rec,MM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{prec,MM,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T) \\
&\equiv \delta_{prec,Mm,\perp\!\!\!\perp}(\hat{Y} \leftarrow B, T)
\end{aligned}
\tag{5.70}
$$

As a consequence of Equation 5.70, which proves the F-Measure to be equivalent to recall and precision, when using macro-averaging together with the rule-independent evaluation strategy, the first one of these evaluation functions is implied to be anti-monotonous according to Definition 3.1. This is, because the variants using the recall metric, respectively the precision metric, are shown to meet said definition in Section 5.2.2.4 and Section 5.2.1.4. Furthermore, as the use of macro-averaging is shown to be equivalent to the use of example-based averaging, when measuring the rule-independent performance of multi-label head rules according to the precision metric, it follows, that the macro-averaged F-Measure is equivalent to that evaluation strategy as well. The equivalences, which are shown in this section, are independent of the F-Measure's β-parameter. ■

5.2.5 Subset Accuracy

In this section, the subset accuracy metric, as defined in Equation 2.18, is examined in terms of anti-monotonicity, when using the rule-independent evaluation strategy. According to the discussion in Section 2.3.3, when using subset accuracy, the performance of a multi-label head rule is calculated by using example-based averaging per definition. Therefore, other averaging strategies – namely micro-averaging, label-based averaging and macro-averaging – must not be considered at this point. In the following it is proved, that the anti-monotonicity property, which is given in Definition 3.1, is not fulfilled by the subset accuracy metric. The proof is given in form of an exemplary search through the label space, which refutes the definition to be met.

Disproof of anti-monotonicity: Figure 5 shows the search tree, which corresponds to an exhaustive search through the label space, given the examples, which are shown in table 7. It uses the subset accuracy metric together with the rule-dependent evaluation strategy for performance evaluation. The dashed line (- - -) indicates the nodes, which are left out, when pruning the search tree under the assumption, that the anti-monotonicity property, according to Definition 3.1, is met by the used evaluation function. In Figure 5, increases of the measured performance are indicated by using green arrows (→). If the performance decreases as result of adding an additional label attribute to the head, which is represented by the preceding node, red arrows (→) are used accordingly. If adding a label attributes does not affect the performance of a multi-label head rule, this is indicated by using black arrows (→).

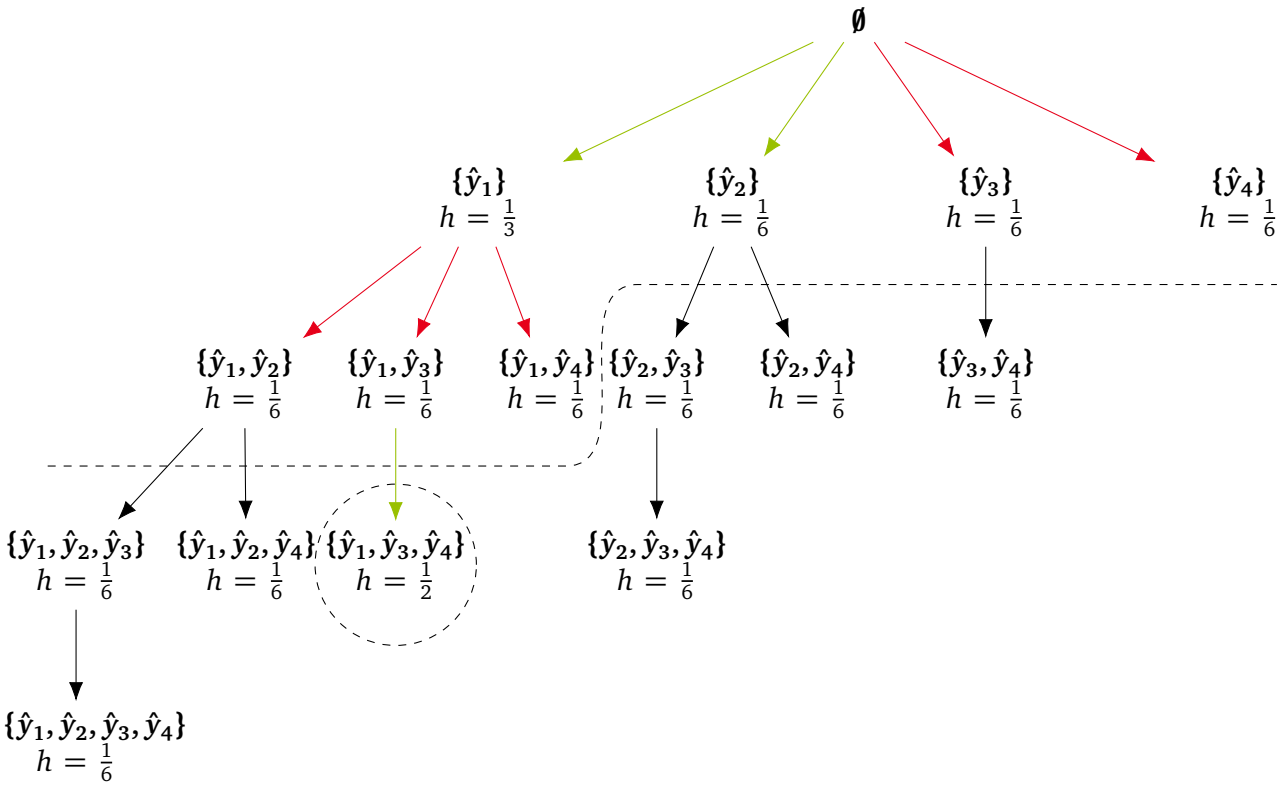

Figure 5: Search through the label space for finding the best multi-label rule head given the examples in Table 7 and using subset accuracy, together with the rule-dependent evaluation strategy, for performance evaluation. The dashed line (- - -) indicates the label combinations, which must not be considered, when pruning the search according to the anti-monotonicity property given in Definition 3.1.

Figure 5 is based on the exemplary label vectors, which are shown in Table 7. Similar to earlier examples, which are given in this work, one half of the examples is assumed to be covered by a rule, whereas the others are not.

		λ_1	λ_2	λ_3	λ_4
Not covered	Y_1	0	1	1	0
	Y_2	1	1	1	1
	Y_3	0	0	0	0
Covered	Y_4	1	0	1	1
	Y_5	1	0	1	1
	Y_6	1	0	0	0

Table 7: Exemplary label vectors of training examples used by Figure 5 and given the label space $\mathbb{L} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. Some examples are assumed to be covered by a given body, some are not.

As it can be seen in Figure 5, pruning the search prevents the label combination $\{\hat{y}_1, \hat{y}_3, \hat{y}_4\}$, which reaches the best performance $\frac{1}{2}$, from being found. Instead, a pruned search results in the head $\{\hat{y}_1\}$ to be considered best, although it only reaches a performance of $\frac{1}{3}$. This is, because the measured performance decreases, when adding the label attribute $\hat{y}_3$ to the head $\{\hat{y}_1\}$. According to the definition of anti-monotonicity, label combinations, which result from adding additional labels, must not be considered, as it is assumed, that the performances of the resulting multi-label head rules cannot reach the best possible performance. However, the given counterexample reveals, that this is not always guaranteed, when using the subset accuracy metric together with the rule-independent evaluation strategy. As adding the label attribute $\hat{y}_4$ to the head $\{\hat{y}_1, \hat{y}_3\}$ not only causes the performance to increase, but also results in the best possible performance among all possible heads, Definition 3.1 is proved to not be met. Consequently, the subset accuracy metric must be considered to not be anti-monotonous, when used together with the rule-independent evaluation strategy. ■

6 Evaluation

In order to statistically evaluate the outcome of the algorithm, which is proposed in the present work, an implementation of the algorithm has been tested on different multi-label data sets. The implementation utilizes the SeCo-framework[1] for rule learning, which has been developed at Technische Universität Darmstadt [Janssen and Fürnkranz, 2010, Janssen and Zopf, 2012], and reuses parts of the implementation, which has been elaborated by Loza Mencía and Janssen [2015] as part of their work. The implementation is meant to be a proof-of-concept, rather than focusing on high-performance computations, for which reason it is not applicable on very large data sets. The data sets, which have been chosen for being used in the statistical experiments, are listed in Table 8 below. All of these multi-label data sets are provided for free use by the developers of Mulan[2] – a Java library for multi-label learning [Mulan Development Team, 2016]. The selection of data sets, which is shown in Table 8, consists of data sets from different domains and includes data sets with nominal attributes, as well as with numerical attributes.

Name	Domain	Instances	Nominal	Numeric	Labels	Cardinality	Density	Distinct
Medical	Text	978	1449	0	45	1.245	0.028	94
Emotions	Music	593	0	72	6	1.869	0.311	27
Genbase	Biology	662	1186	0	27	1.252	0.046	32
Scene	Image	2407	0	294	6	1.074	0.179	15
Birds	Audio	645	2	258	19	1.014	0.053	133

Table 8: Characteristics of the data sets, which are used for the statistical evaluation of rule learning algorithms [Mulan Development Team, 2016]. The columns from left to right specify the name of the datasets, the domain of the input instances, the number of instances, the number of nominal and numeric features, the total number of unique labels, the average number of labels per instance (cardinality), the average percentage of relevant labels (label density) and the number of distinct labelsets in the data (cf. [Loza Mencía and Janssen, 2015, Table 4]).

In order to train different rule learning algorithms on the selected data sets and to evaluate the learned models on a test data set afterwards, the examples of the data sets, which are shown in Table 8, have been separated into distinguished training and test examples beforehand. For separating the data sets into training and test data, a ratio of 2:1 has been used. For all of the selected data sets, pre-separated variants according to said ratio are available [Mulan Development Team, 2016].

Based on the data sets, which are shown in Table 8, different variants of rule learning algorithms have been studied. In the remainder of this chapter, the outcome of these variants is compared to each other in terms of predictive performance and characteristics of the learned models. One difference of the tested rule learners corresponds to the heuristic, which is used for measuring the performance of candidate rules during the rule induction process. Except for the recall metric, all heuristics, which are given in Section 2.3.3, have been considered. Recall has not been used, because it is expected to result in a bad predictive performance, when used for selecting candidate rules. This is, because it does not penalize wrong predictions, which assess irrelevant labels as relevant. As a result, too many labels are expected to be predicted as relevant, when using that metric. However, the recall metric has an influence on measurements, which are based on the F-Measure, as it trades off between precision and recall. For variants of rule learners, which use the F-Measure, the β-parameter has been set to 0.5, This results in the measurement to be more precision-oriented. The decision for introducing a bias towards precision has been made in order to prevent the prediction of too many labels as relevant with respect to the characteristics of the recall metric as discussed previously.

[1] More information about the SeCo-framework for rule learning can be found online at `http://www.ke.tu-darmstadt.de/resources/SeCo`.

[2] The website of the Java library "Mulan" is available at `http://mulan.sourceforge.net`.

One aim of this chapter is to compare the algorithm, which is discussed in the present work, to other multi-label classification approaches. For this reason, a rule learner, which uses the binary relevance method, and the algorithm for learning single-label head rules by Loza Mencía and Janssen [2015], have been tested as well. The following list provides an overview of the titles, which are used for referring to the different variants in the following, as well as a description of the respective rule learning approaches:

- **"BR":** This variant corresponds to the binary relevance problem transformation method. It is based on transforming the original multi-label classification task into multiple single-class classification problems (cf. Section 2.1.1). For solving the individual subproblems, a separate-and-conquer rule learning algorithm, as provided by the SeCo-framework, has been used. The separate-and-conquer algorithm utilizes a top-down search.

- **"Single":** The separate-and-conquer algorithm for learning single-label head rules, which has been proposed by Loza Mencía and Janssen [2015] (cf. Section 2.2.3). For reasons of computational performance, the authors use a variant of said algorithm, which internally uses JRip for the induction of rules, for the statistical evaluations, which are part of their work. JRip is a rule learner for solving single-class classification problems, based on the famous C4.5 algorithm [Quinlan, 2014]. It is provided for free use as part of the Weka[3] machine learning software. Besides its computational efficiency, the use of JRip enables to post-process the learned rules, which tends to be beneficial in terms of predictive performance. Mencía and Janssen's multi-label classification algorithm is able to use JRip, because it considers only one label at once. However, in order to be able to induce multi-label head rules, the algorithm, which is proposed in this work, cannot use JRip. Because of this, – in order to ensure a fair comparison of predictive performances – a variant of Mencía and Janssen's separate-and-conquer algorithm, which uses a top-down search based on the SeCo-framework, rather than JRip, has been used for the statistical evaluations, which are discussed at this point.

- **"Multi":** The title "Multi" is used in the following to refer to the separate-and-conquer algorithm for learning multi-label head rules as proposed in the work at hand (cf. Chapter 4). In order to measure the performance of individual rules during the rule induction process, different averaging strategies can be used. In the following, the used averaging strategy is indicated by using the subscript notation Multi_{mm} in case of micro-averaging, Multi_{mM} for label-based averaging and Multi_{Mm}, respectively Multi_{MM}, for example-based or macro-averaging. Furthermore, when using an algorithm, which is able to induce multi-label head rules, using the subset accuracy metric (cf. Equation 2.18) for selecting candidate rules, is a viable option. This is, because when using such algorithm, all available labels are taken into account whenever inducing a new rule. When using the binary relevance method or an algorithm for learning single-label head rules instead, only one label is considered at once. In such case, subset accuracy is equivalent to the example-based hamming accuracy metric (cf. Equation 2.14), which measures the percentage of correctly classified labels among all labels per instance.

By default, the separate-and-conquer algorithm by Loza Mencía and Janssen [2015] only induces rules, which predict the presence of labels (when using the target $G = \{1\}$, cf. Algorithm 4). In addition, it also offers the possibility to learn rules, which predict the absence of labels (when using the targets $G = \{0,1\}$). As the algorithm for learning multi-label head rules, which is proposed in this work, is based on said algorithm, it also provides the possibility to induce both types of rules (cf. Algorithm 13 and Algorithm 14). In the following, the symbol + is used to denominate approaches, which only predict the presence of labels, whereas the symbol ± denotes variants, which also take rules, that predict the absence of labels, into consideration.

[3] The Weka machine learning software can be downloaded at `https://sourceforge.net/projects/weka`.

The operation of both multi-label algorithms – the one for learning single-label head rules, as well as the one for learning multi-label head rules – heavily depends on the strategy, which is used for re-inserting training examples into the training process. In addition to using variants, which do not re-include fully-covered training examples, variants, which provide fully covered examples to later iterations of the respective separate-and-conquer algorithm, have been tested. For variants, which are based on an re-inserting fully covered examples, the τ-parameter was set to 0.01. In the remainder of this chapter, these variants are identified by using the keyword *stop*.

6.1 Predictive Performance

One important goal of the statistical evaluations, which are discussed in this section, was to investigate the predictive performance of different multi-label classification approaches. To be able to compare the individual approaches with each other, their performances according to the following evaluation metrics have been obtained:

- **"Hamm. Acc.":** This metric corresponds to hamming accuracy, as introduced in Equation 2.14. The hamming accuracy of a learned model has been calculated by obtaining the predictive performances, the model reaches for each example of a test set, and averaging the results example-wise.
- **"Subset Acc.":** According to Equation 2.18, the subset accuracy of a model corresponds to the percentage of test examples, whose label vectors have been predicted perfectly.
- **"Ex.-based Prec.":** This corresponds to the example-based precision metric. According to Equation 2.12, for each test example, the percentage of correctly predicted relevant labels among all labels, which are predicted as relevant, have been calculated. Finally, the obtained performances have been averaged example-wise.
- **"Ex.-based Rec.":** This metric corresponds to the example-based recall. According to Equation 2.13, for each test example, it measures the fraction of predicted relevant labels among all relevant labels. The performances, which have been obtained for each test example, have been averaged example-wise.
- **"Ex.-based F1":** The F1-Measure, as introduced in Equation 2.17, trades of between precision and recall. Because the β-parameter is set to 1, both metrics are weighted equally. When using the example-based variant of the F1-Measure, a performance is obtained per test example at first. Afterwards, the results are averaged example-wise.
- **"Mi. Prec.":** This corresponds to the micro-averaged variant of the precision metric. According to the definition of micro-averaging, the evaluation metric has been applied to the true positives, false positives, true negatives and false negatives, which have been aggregated over the whole test data set (cf. Equation 2.8).
- **"Mi. Rec.":** When using this evaluation metric, according to the definition of micro-averaging, the predictive outcome of a model on a test data set is aggregated at first. Afterwards, the recall metric is applied to the aggregated information.
- **"Mi. F1":** This corresponds to the micro-averaged variant of the F1-Measure.

The performances of the tested classification approaches, according to the evaluation metrics, which are discussed above, are given in Appendix A. The statistical experiments have been carried out with help of the "Lichtenberg" high performance computer at Technische Universität Darmstadt[4]. For variants of the approaches Multi+ and Multi±, which use the same evaluation function and averaging strategies and which can be considered to be equivalent according to the examinations in Chapter 5, only one representative has been tested.

[4] For information about the "Lichtenberg" high performance computer, refer to `http://www.hhlr.tu-darmstadt.de`.

In the following, the predictive performances, which are shown in Table 12, 13, 14, 15 and 16, are summarized and analyzed. Each of these tables corresponds to the results, which have been obtained on one of the data sets MEDICAL, EMOTIONS, GENBASE, BIRDS and SCENE. The approaches Multi+ and Multi± have been configured to use a rule-dependent evaluation strategy for candidate selection during the rule induction process (cf. Chapter 3). Variants of said approaches, which used the rule-independent evaluation strategy for inducing multi-label head rules, are discussed separately in the course of this section.

- On most of the considered data sets, the binary relevance algorithm performs moderately. When compared to other approaches, which use the same evaluation function for candidate selection, it never reaches the lowest performance. However, it is almost always outperformed by one or several variants of the algorithms Single+/± or Multi+/±, which are able to exploit label dependencies. Only on the data set GENBASE – when using the F-Measure or hamming accuracy – the BR approach is able to outdo its competitors. This could be, because said data set contains only very weak label dependencies [Loza Mencía and Janssen, 2015].

- In the majority of cases, the approach Single± reaches a better predictive performance, than its counterpart Single+, which does not induce rules for predicting irrelevant labels. The available statistics do not reveal a clear tendency, whether the use of stopping rules is beneficial or disadvantageous for these algorithms. In many cases, when using the variant Single±, the performance of the learned model does not even change, depending on whether stopping rules are used, or not. The variants of the approach Single± often rank among the best rated algorithms. Although they do not work well with the hamming accuracy metric on the data sets MEDICAL and EMOTIONS, these variants are often even able to reach the best performance among all approaches, if the correct heuristic is used. Especially on the data set BIRDS, they reach very good results, when compared to their competitors.

- When using example-based averaging or macro-averaging together with the Multi+/± approaches, the resulting performance is almost always worse than when using the micro-averaging or label-based counterpart. This does not include the use of the hamming accuracy metric, for which all averaging variants are equivalent. Also, when using the precision metric on the data set EMOTIONS, the example-based and macro-averaged variants of the Multi+$_{stop}$ approach are ranked higher than the micro-averaged and macro-averaged variants.

- On average, the algorithm, which is proposed in this work, seems to benefit from learning rules, which predict irrelevant labels. This is based on the observation, that the best Multi± approach outperforms the best Multi+ variant most of the time. However, when using the Multi±$_{Mm}$ and Multi±$_{MM}$ approaches, the performances of the learned model, according to the precision, recall and F1 metric, often evaluate to 0%. This indicates, that only associations of irrelevant labels are modeled in these cases. As no rules predict the presence of labels, such models are useless.

- Based on the available statistics, it is hard to tell, if the use of stopping rules has a positive impact on the resulting performances of the Multi+/± approaches, or not. The respective outcomes heavily depend on the used evaluation function and data set.

- Except for the data set GENBASE, at least one variant of the proposed algorithm is always able to outperform the BR approach, regardless of the used evaluation function. In many cases – depending on the data set and used evaluation function –, variants of the algorithm even allow to reach the highest rank among all approaches. Most notable, by applying the approaches Multi±$_{stop,mm}$ or Multi±$_{stop,mM}$ on the data set EMOTIONS, the highest performances can be reached, regardless of the evaluation function, which is used for selecting candidate rules.

– Unlike the binary relevance method and the multi-label classification approach by Loza Mencía and Janssen, the algorithm for learning multi-label head rules, which is proposed in the present work, is able to use the subsect accuracy metric for selecting candidate rules during the rule induction process. As the available statistics reveal, the performances, which result from using said metric, can most of the time compete with the results of the other approaches. Only on the data set SCENE there is significant difference between the performances, which can be reached by using the subset accuracy metric, and those of the highest ranked approaches.

In addition to using a rule-dependent evaluation, the performance of the algorithm for learning multi-label head rules has also been investigated, when using the rule-independent evaluation strategy. The results of these experiments are shown in Table 17, 18, 19, 20 and 21 of Appendix A.

– When compared to the performances of models, which have been learned by using the rule-dependent evaluation strategy, the predictive performance seems to suffer from using rule-independent evaluations. Regardless of the used data set, variants of the algorithm, which rely on the rule-independent evaluation strategy for candidate selection, almost always reach worse performances on average, than their rule-dependent counterparts. Whereas this is especially evident for the metrics subset accuracy, precision and F1-Measure, the rule-independent approaches often reach a hamming accuracy, which is comparable to those of their rule-dependent counterparts. Furthermore, in many cases, they even reach a higher performance according to the recall metric. This could possibly be an indicator, that using the rule-independent evaluation strategy often results in too many labels being predicted as relevant.

– Except for the data sets EMOTIONS and SCENE, the Multi± approaches, which use the rule-independent evaluation strategy and are able to predict the irrelevance of labels, did not finish in time. This is, because in such case, the algorithm tends to induce rules, which predict fully set label vectors. This prevents searches through the label space from being pruned early and therefore results in a high computational complexity. However, from the statistical results of the approaches, which did finish in time, it can be seen, that precision, recall and F1-Measure often evaluation to 0%. As already discussed, this is an indicator, that only rules, which predict irrelevant labels, are learned in such case.

– According to the examinations in Section 5.2.5, the subset accuracy metric does not meet the definition of anti-monotonicity, when using the rule-dependent evaluation strategy. Because of this, it does not allow to prune searches through the label space and therefore it has not been used in the statistical experiments.

6.2 Characteristics of Learned Models

In addition to the comparison of the predictive performances of different multi-label classification approaches, the models, which have been learned by the separate-and-conquer algorithm by Loza Mencía and Janssen [2015], as well as by the algorithm for learning multi-label head rules, are compared to each other in this section. The rules, which have been learned by using the binary relevance method, are not considered at this point. This is, because such approaches are based on training multiple single-class classifiers, rather than learning single decision list. In order to statistically investigate the characteristics of learned models, the following properties have been taken in consideration. The statistics, which have been obtained with respect to these properties, are shown in Appendix B.

- **"# Rules":** The number of rules a model consists of. This does not include stopping rules.
- **"# Stopping Rules":** The number of stopping rules, which have been created during the rule induction process.

- **"# Label Conditions":** The number of label conditions among all induced rules of a model.
- **"% Full Label-dependent":** The percentage of rules, which exclusively contain label conditions in their bodies.
- **"% Partially Label-dependent":** The percentage of rules, which contain label conditions, as well as attribute conditions, in their bodies.
- **"% Not Label-dependent":** The percentage of rules, whose bodies do not contain any label conditions.
- **"# Multi-Label Head Rules":** The number multi-label head rules, which are contained by the learned model.
- **"% Multi-Label Head Rules":** The percentage of multi-label head rules among all induced rules.
- **"Avg. # Labels per Head":** The number of label attributes, the induced rules contain in their heads on average.

In Table 22, 23, 24, 25 and 26, the characteristics of the models, which have been learned by using the rule-dependent evaluation strategy, are shown. Each of these tables corresponds to one of the data sets MEDICAL, EMOTIONS, GENBASE, BIRDS and SCENE. Based on the obtained data, the following conclusions can be made:

- When using example-based averaging or macro-averaging together with the F-Measure or the precision metric for measuring the performance of candidate rules, the Multi+/± approaches on average induce far less rules, than when using micro-averaging or label-based averaging. As mentioned in the previous section, these approaches also tend to result in poor predictive performance. Probably this is, because the few rules, which are learned by said approaches, overgeneralize on the training data and therefore are not able to model label associations in a very differentiated way. Furthermore, if only few rules are learned by an algorithm, these rules are unlikely to contain label conditions in their bodies.
- On average, the percentage of rules, which either partially or fully depend on label conditions, seems to increase when using the Single± approaches, rather than the variants of the Single+ approach. This does also apply to the Multi± variants, which tendentially result in more label-dependent rules being learned, than when using the Multi+ approaches. The amounts of label-dependent rules, which have been induced by using the algorithm by Loza Mencía and Janssen, respectively by the algorithm for learning multi-label head rules, are close to each other for the most time, if micro- or label-based averaging is used by the latter algorithm. However, when using the precision metric, the label-dependent rules, which are induced by the first of both algorithms, outnumber those that are learned by the latter one. A possible explanation of this particularity could be, that the latter algorithm results in many multi-label head rules being learned in these cases. Because multi-label head rules are also able to model label correlations, they might be preferred over label-dependent rules.
- In general, significantly more multi-label head rules are induced by the Multi+/± approaches, when using the precision metric, rather than the F-Measure, hamming accuracy or subset accuracy, for selecting candidate rules. Also, the learned multi-label head rules tend to consist of more label attributes in those cases. The reasons for this behavior are discussed more detailed in the following.

As the statistical evaluation of the learned models revealed, only few multi-label head rules are induced when using any evaluation metric other than the precision metric together with a rule-dependent evaluation. In the following, this phenomenon is explained by giving an example. The example is based on the

exemplary label vectors of fictional training examples, which are shown in Table 9. Two of the examples, which are given in said table, are assumed to be covered by a rule, whereas the remaining examples are assumed to be uncovered. According to Table 9, the labels λ_1 and λ_2 are both associated with the covered examples, whereas the other labels are irrelevant. It seems obvious, that a multi-label head, which predicts both, the label λ_1, as well as the label λ_2 as relevant, models the label associations of the covered examples best. However, only if the precision metric is used for measuring the performance of potential heads, such a multi-label head is chosen. Using a different evaluation function results in a single-label head to be chosen instead.

		λ_1	λ_2	λ_3	λ_4
Not covered	Y_1	0	0	1	0
	Y_2	0	0	1	1
	Y_3	0	1	1	0
	Y_4	0	0	1	1
Covered	Y_5	1	1	0	0
	Y_6	1	1	0	0

Table 9: Exemplary label vectors of training examples referred to in Table 10 and given the label space $\mathbb{L} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. Some examples are covered by a potential rule's body, some are not.

In Table 10, the performances of the single-label heads $\{\hat{y}_1\}$ and $\{\hat{y}_2\}$, as well as of the multi-label head $\{\hat{y}_1, \hat{y}_2\}$, according to different evaluation functions, are shown. Only if the precision metric is used, the multi-label head $\{\hat{y}_1, \hat{y}_2\}$ is preferred over the single-label head $\{\hat{y}_1\}$. The given performances are calculated by using micro-averaging and the rule-dependent evaluation strategy. However, when using a different averaging strategy, the performance evaluations would result in a similar outcome.

	Precision	Recall	Hamming Accuracy	F-Measure ($\beta = 0.5$)	Subset Accuracy
$\{\hat{y}_1\}$	1	(1)	(1)	(1)	(1)
$\{\hat{y}_2\}$	1	$\frac{5}{6}$	$\frac{2}{3}$	$\frac{10}{11}$	$\frac{5}{6}$
$\{\hat{y}_1, \hat{y}_2\}$	(1)	$\frac{11}{12}$	$\frac{4}{5}$	$\frac{20}{21}$	$\frac{5}{6}$

Table 10: Performance of the multi-label heads $\{\hat{y}_1\}$, $\{\hat{y}_2\}$ and $\{\hat{y}_1, \hat{y}_2\}$ according to different evaluation functions and given the label vectors in Table 9. In this example micro-averaging is used together with the rule-dependent evaluation strategy. The performance of the best rated head according to each evaluation function is circled.

As Table 10 reveals, the use of recall, hamming accuracy, F-Measure or subset accuracy does not result in a multi-label head rule to be chosen, because the single-label head rule $\{\hat{y}_1\}$ is rated higher in all of these cases. This is, because all of these evaluation functions take true negatives or false negatives into account and therefore depend on the uncovered examples. The label λ_2 is set in the label vector Y_3, but it is never predicted as relevant, because the given rule does not cover the corresponding example. The label is therefore counted as a false negative. When using the rule-dependent evaluation strategy, this causes the performance of a rule, which predicts label λ_2 as relevant, to decrease in comparison to a rule, which only predicts label λ_1. This is, because, when only predicting the label λ_1, the predictions for that label are considered as perfect. According to the rule-dependent evaluation strategy, the label λ_2 is not taken into account at all in such case. If the label λ_2 is predicted in addition, it is taken into account by the performance evaluation and causes the overall performance of the rule to suffer, because the predictions are not considered as perfect in case of the label vector Y_3.

In the following, the characteristics of models, which have been learned by using the rule-independent evaluation strategy, are analyzed. This corresponds to the statistics, which are shown in Table 27, 28, 29, 30 and 31 of Appendix B.

- The models of approaches, which utilize the rule-independent evaluation strategy, tend to consist of more multi-label head rules than those of corresponding approaches using a rule-dependent evaluation. The number of induced rules – and therefore the chance of multi-label head rules being learned – strongly depends on the used averaging strategy. When using micro-averaging or label-based averaging, tendentially more rules are learned, than when using example-based averaging or macro averaging. This does not apply to the hamming accuracy metric, because all averaging strategies are equivalent in that case. Moreover, the use of stopping rules seems to have an impact on the size of the learned models as well. Approaches, which make use of stopping rules, often result in more rules being learned, than corresponding approaches, which do not use stopping rules.
- When using the rule-independent evaluation strategy, while allowing to model the associations of irrelevant labels, all induced rules are multi-label head rules. Furthermore, all of these rules predict a fully set label vector, i.e. for all available labels a prediction is made. When micro-averaging or label-based averaging is used together with the precision metric, a large number of rules is induced. This allows to conclude, that the Multi± approaches tend to overfitting in such case. This is, because the rules only cover few examples and predict their full label vectors. When using different averaging strategies or evaluation functions, only very few rules are learned. Such rules are prone to overgeneralize, as they cover a lot of examples and predict a full label vector, instead of differentiating between the labels, which are associated with individual examples.

7 Conclusion

As the conclusion of this work, a summary of its most important contributions is given in this last chapter. On the one hand, this includes an overview of the previous chapters' contents. On the other hand, possible improvements of the proposed algorithm and further investigations, which are not part of this work, are pointed out as well.

7.1 Summary

In this work, a separate-and-conquer rule learning algorithm, which is able to induce multi-label head rules, was proposed. As it is based on the separate-and-conquer algorithm for multi-label classification by Loza Mencía and Janssen [2015], it reuses some aspects, which have been elaborated as part of the original algorithm's publication. Such as the original algorithm, the approach, which was presented in this work, optionally enables to use stopping rules and is able to model associations of irrelevant labels in addition to those of relevant labels. In order to be able to implement the search for multi-label head rules in an efficient way, the properties of anti-monotonous and decomposable evaluation functions were formally defined. As it was illustrated in this work, by exploiting these properties, searches through the label space can be pruned by leaving out the evaluation of unpromising label combinations. Furthermore, common metrics for measuring the performance of multi-label head rules – namely precision, recall, hamming accuracy, subset accuracy and the F-Measure – were examined in terms of anti-monotonicity and decomposability. As the examinations revealed, most of these metrics are suited for being used for pruned searches. When using a rule-dependent evaluation strategy, most metrics meet the definition of decomposability. No deep searches through the label space are required in such cases at all. When using a rule-independent evaluation instead, most metrics meet the definition of anti-monotonicity, which enables to prune searches less extensively. An overview of whether the considered metrics fulfill the definitions of anti-monotonicity and decomposability, according to the examinations, which were part of this work, is given in Table 7.2. By statistically evaluating the outcome of the proposed algorithm on different multi-label data set, its predictive performance was compared to those of other multi-label classification approaches. As the experiments revealed, the algorithm is able to outperform the popular binary relevance method, if correlations between labels are given in a data set. This is, because the algorithm is able to model such correlations by using label-dependent rules, as well as by rules, which contain multiple label predictions in their heads. It was further shown, that the presented algorithm can compete with the algorithm for learning single-label head rules by Loza Menćia and Janssen in terms of predictive performance. Especially when using the rule-dependent evaluation strategy together with micro-averaging or example-based averaging, the algorithm results in reliable predictions. Depending on the data set, as well as on the used heuristic for selecting candidate rules, the original algorithm could even be outperformed in some cases.

7.2 Future Work

Besides the evaluation metrics, which were considered in this work, other heuristics – e.g. the Jaccard metric [Gjorgjioski et al., 2011] – are suited to be used by the proposed algorithm as well. In order to use different metrics than those, which were considered in this work, they must be examined in terms of anti-monotonicity, respectively decomposability, beforehand. This is necessary to ensure, that they are suited for being used for pruned searches through the label space. However, the examination of additional metrics is left for future work, if necessary.

During the elaboration of this work, there were considerations to utilize a beam-search for refining the conditions of potential rules. This idea was motivated by the fact, that in each refinement step, the proposed algorithm only chooses one single rule. At first, only rules, which contain a single condition in their body, are considered. Among these rules, the one, which reaches the best performance, is chosen. Afterwards, all possible refinements, which result from adding an additional condition to the chosen rule's body, are evaluated. Among these refinements, only the highest rated rule is chosen again. This

process continues until no refinements result in a higher performance being reached anymore. In theory, the rules which are discarded during the refinement process, might outperform the rule, which is finally chosen by the algorithm, because their performances might increase when adding additional conditions to their body. As refining all potential rules would require to perform an exhaustive with exponential computational complexity, this is not feasible in practice. As an alternative, a beam-search would allow to keep track of the most β promising rules and refine all of them (where $\beta \in \mathbb{N}$ is a beam width greater or equal to 1). However, although a beam search was implemented, the effects of using such approach were not further investigated. As a possible topic of a future work, it might be interesting to evaluate, whether the use of a beam-search has the potential to increase the algorithm's predictive performance and therefore justifies the negative impact it has in terms of computational complexity.

As the empirical studies on different data sets have revealed, when selecting candidate rules based on a heuristic other than the precision metric, multi-label head rules are unlikely to be induced. This is, because these metrics take true negatives and false negatives into account and therefore uncovered examples have an impact on the measured performance. In order to introduce a bias towards learning multi-label head rules, when using such heuristics, it would be possible to weight the performances of rules, depending on the number of labels they predict. Rules, which predict more labels should be rated better than those, that predict less labels. For example, this could be achieved by introducing a parameter $k \in \mathbb{R}$ (e.g. $k = 1.1$), which affects the performance h of a multi-label head rule $\hat{Y} \leftarrow B$ according to the following equation. However, as the use of such weights directly affects the performances of rules, the effects on the anti-monotonicity and decomposability properties of individual evaluation functions must further be investigated.

$$h = \delta(\hat{Y} \leftarrow B, T) \cdot k^{|\hat{Y}|-1} \text{ , with } k \geq 1$$

When using a rule-dependent evaluation, the best multi-label head of a rule is derived from the performances of single-label head rules. The labels, which are predicted by single-label head rules, which reach the highest performance, are combined in order to make up the best multi-label head. By default, the predictions of single-label head rules are only combined, if those rules reach the exact same performance. By relaxing this constraint, it would be possible to introduce a bias towards the induction of multi-label head rules. This could be achieved by introducing a parameter $\epsilon \in \mathbb{R}$ (e.g. $\epsilon = 0.05$), which specifies a tolerance limit. Even if the performance h of a single-head rule is less than the best performance h_{max}, its prediction is taken into account for making up the best multi-label head rule, if $h_{best} - h \leq \epsilon$. Investigating the effects of introducing a bias towards the induction of multi-label head rules on the predictive performance and learned models of the proposed algorithm is left for future work.

Evaluation Function	Evaluation Strategy	Averaging Strategy	Anti-Monotonicity / Decomposability	
Precision	Rule-dependent	Micro-Averaging	Yes / Yes	1
		Label-based Averaging	Yes / Yes	1
		Example-based Averaging	Yes / Yes	2
		Macro-Averaging	Yes / Yes	2
	Rule-independent	Micro-Averaging	Yes / -	3
		Label-based Averaging	Yes / -	3
		Example-based Averaging	Yes / -	4
		Macro-Averaging	Yes / -	4
Recall	Rule-dependent	Micro-Averaging	Yes / Yes	
		Label-based Averaging	Yes / Yes	
		Example-based Averaging	- / -	
		Macro-Averaging	Yes / Yes	2
	Rule-independent	Micro-Averaging	Yes / -	
		Label-based Averaging	Yes / -	
		Example-based Averaging	- / -	
		Macro-Averaging	Yes / -	4
Hamming Accuracy	Rule-dependent	Micro-Averaging	Yes / Yes	5
		Label-based Averaging	Yes / Yes	5
		Example-based Averaging	Yes / Yes	5
		Macro-Averaging	Yes / Yes	5
	Rule-independent	Micro-Averaging	Yes / -	6
		Label-based Averaging	Yes / -	6
		Example-based Averaging	Yes / -	6
		Macro-Averaging	Yes / -	6
F-Measure	Rule-dependent	Micro-Averaging	Yes / Yes	
		Label-based Averaging	Yes / Yes	
		Example-based Averaging	Yes / Yes	
		Macro-Averaging	Yes / Yes	2
	Rule-independent	Micro-Averaging	Yes / -	
		Label-based Averaging	Yes / -	
		Example-based Averaging	Yes / -	
		Macro-Averaging	Yes / -	4
Subset Accuracy	Rule-dependent	Example-based Averaging	Yes / -	
	Rule-independent		- / -	

Table 11: Anti-monotonicity and decomposability of selected evaluation functions, regarding different averaging and evaluation strategies. The numbers at the right indicate equivalent variants.

A Results of Performance Evaluations

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
F-Measure									
BR	98.60% 5	55.81% 5	71.30% 5	76.07% 11	71.65% 6	74.38% 3	75.13% 11	74.75% 5	5
Single+$_{stop}$	97.34% 13	22.17% 13	52.71% 13	75.45% 12	59.96% 13	51.26% 13	73.75% 15	60.48% 13	15
Single+	98.17% 10	45.58% 10	67.65% 10	87.88% 5	73.47% 4	61.97% 10	87.38% 5	72.51% 8	6
Multi+$_{stop,mm}$	98.45% 8.5	53.80% 8.5	68.70% 8.5	74.11% 16.5	69.45% 11.5	71.15% 8.5	73.38% 16.5	72.25% 9.5	11.5
Multi+$_{stop,mM}$	98.45% 8.5	53.80% 8.5	68.70% 8.5	74.11% 16.5	69.45% 11.5	71.15% 8.5	73.38% 16.5	72.25% 9.5	11.5
Multi+$_{stop,Mm}$	18.77% 20.5	0.00% 17.5	3.24% 16.5	98.14% 1.5	6.23% 16.5	3.21% 16.5	95.75% 1.5	6.22% 16.5	16.5
Multi+$_{stop,MM}$	18.77% 20.5	0.00% 17.5	3.24% 16.5	98.14% 1.5	6.23% 16.5	3.21% 16.5	95.75% 1.5	6.22% 16.5	16.5
Multi+$_{mm}$	98.00% 11.5	45.12% 11.5	64.89% 11.5	82.58% 6.5	69.66% 9.5	60.04% 11.5	81.88% 6.5	69.28% 11.5	9.5
Multi+$_{mM}$	98.00% 11.5	45.12% 11.5	64.89% 11.5	82.58% 6.5	69.66% 9.5	60.04% 11.5	81.88% 6.5	69.28% 11.5	9.5
Multi+$_{Mm}$	56.02% 18.5	0.00% 17.5	5.56% 14.5	93.70% 3.5	10.41% 14.5	5.55% 14.5	93.38% 3.5	10.48% 14.5	13.5
Multi+$_{MM}$	56.02% 18.5	0.00% 17.5	5.56% 14.5	93.70% 3.5	10.41% 14.5	5.55% 14.5	93.38% 3.5	10.48% 14.5	13.5
Single±$_{stop}$	98.79% 1	62.79% 1	75.45% 1	78.19% 10	75.26% 1	78.21% 2	77.63% 10	77.92% 1	1
Single±	98.78% 2	59.69% 4	72.51% 4	74.96% 15	72.23% 5	79.92% 1	74.63% 12	77.18% 2	4
Multi±$_{stop,mm}$	98.63% 3.5	59.84% 2.5	73.68% 2.5	79.84% 8.5	74.89% 2.5	73.32% 6.5	79.00% 8.5	76.05% 3.5	2.5
Multi±$_{stop,mM}$	98.63% 3.5	59.84% 2.5	73.68% 2.5	79.84% 8.5	74.89% 2.5	73.32% 6.5	79.00% 8.5	76.05% 3.5	2.5
Multi±$_{stop,Mm}$	97.24% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	20.5
Multi±$_{stop,MM}$	97.24% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	18.5
Multi±$_{mm}$	98.57% 6.5	55.66% 6.5	70.17% 6.5	75.19% 13.5	70.80% 7.5	73.95% 4.5	74.50% 13.5	74.22% 6.5	7.5
Multi±$_{mM}$	98.57% 6.5	55.66% 6.5	70.17% 6.5	75.19% 13.5	70.80% 7.5	73.95% 4.5	74.50% 13.5	74.22% 6.5	7.5
Multi±$_{Mm}$	97.24% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	20.5
Multi±$_{MM}$	97.24% 14.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	18.5
Hamming Accuracy									
BR	98.48% 15	54.57% 15	68.89% 15	73.80% 11	69.46% 17	72.11% 15	73.38% 11	72.74% 13	17
Single+$_{stop}$	97.58% 20	29.61% 20	56.99% 20	77.93% 5	63.59% 20	54.35% 20	76.50% 6	63.55% 20	20
Single+	97.07% 21	20.00% 21	49.50% 21	77.49% 6	57.84% 21	48.11% 21	77.75% 5	59.44% 21	21
Multi+$_{stop,mm}$	98.60% 10.5	58.29% 2.5	73.32% 2.5	77.11% 8.5	73.27% 2.5	74.17% 10.5	75.75% 8.5	74.95% 2.5	2.5
Multi+$_{stop,mM}$	98.60% 10.5	58.29% 2.5	73.32% 2.5	77.11% 8.5	73.27% 2.5	74.17% 10.5	75.75% 8.5	74.95% 2.5	2.5
Multi+$_{stop,Mm}$	98.60% 10.5	58.29% 2.5	73.32% 2.5	77.11% 8.5	73.27% 2.5	74.17% 10.5	75.75% 8.5	74.95% 2.5	2.5
Multi+$_{stop,MM}$	98.60% 10.5	58.29% 2.5	73.32% 2.5	77.11% 8.5	73.27% 2.5	74.17% 10.5	75.75% 8.5	74.95% 2.5	2.5
Multi+$_{mm}$	98.32% 17.5	48.53% 17.5	67.32% 17.5	78.58% 2.5	70.22% 14.5	66.60% 17.5	78.25% 2.5	71.95% 17.5	14.5
Multi+$_{mM}$	98.32% 17.5	48.53% 17.5	67.32% 17.5	78.58% 2.5	70.22% 14.5	66.60% 17.5	78.25% 2.5	71.95% 17.5	14.5
Multi+$_{Mm}$	98.32% 17.5	48.53% 17.5	67.32% 17.5	78.58% 2.5	70.22% 14.5	66.60% 17.5	78.25% 2.5	71.95% 17.5	14.5
Multi+$_{MM}$	98.32% 17.5	48.53% 17.5	67.32% 17.5	78.58% 2.5	70.22% 14.5	66.60% 17.5	78.25% 2.5	71.95% 17.5	14.5
Single±$_{stop}$	98.50% 13.5	55.35% 13.5	68.94% 13.5	70.90% 20.5	68.30% 18.5	74.02% 13.5	70.50% 20.5	72.22% 14.5	18.5
Single±	98.50% 13.5	55.35% 13.5	68.94% 13.5	70.90% 20.5	68.30% 18.5	74.02% 13.5	70.50% 20.5	72.22% 14.5	18.5
Multi±$_{stop,mm}$	98.64% 4.5	57.52% 8.5	70.63% 8.5	72.87% 15.5	70.23% 8.5	76.82% 4.5	72.50% 15.5	74.60% 8.5	8.5
Multi±$_{stop,mM}$	98.64% 4.5	57.52% 8.5	70.63% 8.5	72.87% 15.5	70.23% 8.5	76.82% 4.5	72.50% 15.5	74.60% 8.5	8.5
Multi±$_{stop,Mm}$	98.64% 4.5	57.52% 8.5	70.63% 8.5	72.87% 15.5	70.23% 8.5	76.82% 4.5	72.50% 15.5	74.60% 8.5	8.5
Multi±$_{stop,MM}$	98.64% 4.5	57.52% 8.5	70.63% 8.5	72.87% 15.5	70.23% 8.5	76.82% 4.5	72.50% 15.5	74.60% 8.5	8.5
Multi±$_{mm}$	98.64% 4.5	57.52% 8.5	70.63% 8.5	72.87% 15.5	70.23% 8.5	76.82% 4.5	72.50% 15.5	74.60% 8.5	8.5
Multi±$_{mM}$	98.64% 4.5	57.52% 8.5	70.63% 8.5	72.87% 15.5	70.23% 8.5	76.82% 4.5	72.50% 15.5	74.60% 8.5	8.5
Multi±$_{Mm}$	98.64% 4.5	57.52% 8.5	70.63% 8.5	72.87% 15.5	70.23% 8.5	76.82% 4.5	72.50% 15.5	74.60% 8.5	8.5
Multi±$_{MM}$	98.64% 4.5	57.52% 8.5	70.63% 8.5	72.87% 15.5	70.23% 8.5	76.82% 4.5	72.50% 15.5	74.60% 8.5	8.5
Precision									
BR	97.65% 7	31.47% 7	49.56% 7	56.25% 15	50.25% 7	57.47% 7	56.25% 14	56.85% 7	7
Single+$_{stop}$	96.91% 14	24.65% 10	45.75% 10	56.72% 14	48.29% 8	45.11% 10	55.88% 15	49.92% 10	11
Single+	95.91% 15	17.98% 11	39.46% 11	67.91% 6	46.33% 11	36.80% 11	67.63% 5	47.67% 11	8
Multi+$_{stop,mm}$	97.22% 12.5	29.77% 8.5	47.63% 8.5	52.07% 16.5	47.71% 9.5	49.52% 8.5	51.75% 16.5	50.61% 8.5	9.5
Multi+$_{stop,mM}$	97.22% 12.5	29.77% 8.5	47.63% 8.5	52.07% 16.5	47.71% 9.5	49.52% 8.5	51.75% 16.5	50.61% 8.5	9.5
Multi+$_{stop,Mm}$	18.77% 20.5	0.00% 17.5	3.24% 16.5	98.14% 1.5	6.23% 16.5	3.21% 16.5	97.75% 1.5	6.22% 16.5	16.5
Multi+$_{stop,MM}$	18.77% 20.5	0.00% 17.5	3.24% 16.5	98.14% 1.5	6.23% 16.5	3.21% 16.5	97.75% 1.5	6.22% 16.5	16.5
Multi+$_{mm}$	95.67% 16.5	14.57% 12.5	38.09% 12.5	65.79% 9.5	44.58% 12.5	34.85% 12.5	65.88% 7.5	45.59% 12.5	12.5
Multi+$_{mM}$	95.67% 16.5	14.57% 12.5	38.09% 12.5	65.79% 9.5	44.58% 12.5	34.85% 12.5	65.88% 7.5	45.59% 12.5	12.5
Multi+$_{Mm}$	56.02% 18.5	0.00% 17.5	5.56% 14.5	93.70% 3.5	10.41% 14.5	5.55% 14.5	93.38% 3.5	10.48% 14.5	14.5
Multi+$_{MM}$	56.02% 18.5	0.00% 17.5	5.56% 14.5	93.70% 3.5	10.41% 14.5	5.55% 14.5	93.38% 3.5	10.48% 14.5	14.5
Single±$_{stop}$	98.34% 1	49.92% 3	66.23% 1	68.86% 5	65.70% 1	70.92% 1	67.38% 6	69.10% 1	1
Single±	98.20% 6	45.74% 6	62.16% 4	65.74% 11	62.14% 4	68.10% 6	65.38% 9	66.71% 4	4
Multi±$_{stop,mm}$	98.26% 2.5	51.47% 1.5	64.55% 2.5	65.87% 7.5	63.69% 2.5	70.29% 2.5	63.88% 10.5	66.93% 2.5	2.5
Multi±$_{stop,mM}$	98.26% 2.5	51.47% 1.5	64.55% 2.5	65.87% 7.5	63.69% 2.5	70.29% 2.5	63.88% 10.5	66.93% 2.5	2.5
Multi±$_{stop,Mm}$	97.24% 9.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{stop,MM}$	97.24% 9.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{mm}$	98.21% 4.5	49.77% 4.5	62.15% 5.5	63.62% 12.5	61.44% 5.5	69.76% 4.5	62.00% 12.5	65.65% 5.5	5.5
Multi±$_{mM}$	98.21% 4.5	49.77% 4.5	62.15% 5.5	63.62% 12.5	61.44% 5.5	69.76% 4.5	62.00% 12.5	65.65% 5.5	5.5
Multi±$_{Mm}$	97.24% 9.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{MM}$	97.24% 9.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Subset Accuracy									
Multi+$_{stop,Mm}$	98.55% 1	58.76% 1	72.12% 1	75.25% 2	72.09% 1	73.42% 1	74.25% 2	73.83% 1	1
Multi+$_{Mm}$	98.24% 2	46.82% 2	65.96% 2	78.50% 1	69.26% 2	65.04% 2	78.38% 1	71.09% 2	2
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	

Table 12: Predictive performance of different multi-label classification approaches on the data set MEDICAL using the rule-dependent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
F-Measure									
BR	72.94% 3	18.81% 7	57.03% 1	53.22% 13	51.89% 5	59.83% 3	54.14% 14	56.84% 5	3
Single+$_{stop}$	65.26% 16	9.41% 12	47.52% 12	59.32% 8	49.68% 13	47.79% 12	59.65% 8	53.07% 13	13
Single+	64.11% 17	10.89% 10	45.25% 13	67.16% 7	50.79% 12	46.85% 13	67.17% 7	55.20% 10	12
Multi+$_{stop,mm}$	71.62% 8.5	14.36% 8.5	55.07% 4.5	55.69% 9.5	51.67% 6.5	57.11% 8.5	55.39% 9.5	56.23% 8.5	8.5
Multi+$_{stop,mM}$	71.62% 8.5	14.36% 8.5	55.07% 4.5	55.69% 9.5	51.67% 6.5	57.11% 8.5	55.39% 9.5	56.23% 8.5	8.5
Multi+$_{stop,Mm}$	33.17% 19.5	0.00% 17.5	33.04% 15.5	100.00% 2.5	48.59% 15.5	33.00% 15.5	100.00% 2.5	49.63% 15.5	15.5
Multi+$_{stop,MM}$	33.17% 19.5	0.00% 17.5	33.04% 15.5	100.00% 2.5	48.59% 15.5	33.00% 15.5	100.00% 2.5	49.63% 15.5	15.5
Multi+$_{mm}$	68.65% 10.5	9.41% 12	53.45% 10.5	70.54% 5.5	56.66% 1.5	51.75% 10.5	70.43% 5.5	59.66% 1.5	4.5
Multi+$_{mM}$	68.65% 10.5	9.41% 12	53.45% 10.5	70.54% 5.5	56.66% 1.5	51.75% 10.5	70.43% 5.5	59.66% 1.5	4.5
Multi+$_{Mm}$	33.17% 19.5	0.00% 17.5	33.04% 15.5	100.00% 2.5	48.59% 15.5	33.00% 15.5	100.00% 2.5	49.63% 15.5	15.5
Multi+$_{MM}$	33.17% 19.5	0.00% 17.5	33.04% 15.5	100.00% 2.5	48.59% 15.5	33.00% 15.5	100.00% 2.5	49.63% 15.5	15.5
Single±$_{stop}$	73.27% 1.5	19.31% 5.5	55.03% 6.5	52.48% 16.5	51.02% 10.5	60.50% 1.5	54.14% 14	57.14% 3.5	6.5
Single±	73.27% 1.5	19.31% 5.5	55.03% 6.5	52.48% 16.5	51.02% 10.5	60.50% 1.5	54.14% 14	57.14% 3.5	6.5
Multi±$_{stop,mm}$	72.52% 4.5	21.29% 1.5	56.19% 2.5	55.45% 11.5	52.79% 3.5	58.92% 4.5	54.64% 11.5	56.70% 6.5	1.5
Multi±$_{stop,mM}$	72.52% 4.5	21.29% 1.5	56.19% 2.5	55.45% 11.5	52.79% 3.5	58.92% 4.5	54.64% 11.5	56.70% 6.5	1.5
Multi±$_{stop,Mm}$	67.08% 13.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	20.5
Multi±$_{stop,MM}$	67.08% 13.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	18.5
Multi±$_{mm}$	71.70% 6.5	20.79% 3.5	54.91% 8.5	53.05% 14.5	51.21% 8.5	57.65% 6.5	52.88% 16.5	55.16% 11.5	10.5
Multi±$_{mM}$	71.70% 6.5	20.79% 3.5	54.91% 8.5	53.05% 14.5	51.21% 8.5	57.65% 6.5	52.88% 16.5	55.16% 11.5	10.5
Multi±$_{Mm}$	67.08% 13.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	20.5
Multi±$_{MM}$	67.08% 12.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	18.5
Hamming Accuracy									
BR	71.53% 11	15.35% 11	53.99% 13	54.62% 11	50.22% 10	57.03% 11	54.89% 11	55.94% 13	13
Single+$_{stop}$	36.55% 21	0.99% 21	34.61% 21	94.47% 1	48.38% 19	33.51% 21	94.24% 1	49.44% 21	21
Single+	59.65% 16	6.93% 20	45.16% 20	76.07% 6	53.55% 5	43.64% 16	77.44% 2	55.83% 14	16
Multi+$_{stop,mm}$	71.29% 13.5	10.89% 15.5	54.24% 10.5	58.75% 8.5	53.01% 7.5	56.06% 13.5	59.15% 8.5	57.56% 2.5	10.5
Multi+$_{stop,mM}$	71.29% 13.5	10.89% 15.5	54.24% 10.5	58.75% 8.5	53.01% 7.5	56.06% 13.5	59.15% 8.5	57.56% 2.5	10.5
Multi+$_{stop,Mm}$	71.29% 13.5	10.89% 15.5	54.24% 10.5	58.75% 8.5	53.01% 7.5	56.06% 13.5	59.15% 8.5	57.56% 2.5	10.5
Multi+$_{stop,MM}$	71.29% 13.5	10.89% 15.5	54.24% 10.5	58.75% 8.5	53.01% 7.5	56.06% 13.5	59.15% 8.5	57.56% 2.5	10.5
Multi+$_{mm}$	57.26% 18.5	10.89% 15.5	46.21% 17.5	77.39% 3.5	54.09% 2.5	41.90% 18.5	77.19% 4.5	54.32% 18.5	16
Multi+$_{mM}$	57.26% 18.5	10.89% 15.5	46.21% 17.5	77.39% 3.5	54.09% 2.5	41.90% 18.5	77.19% 4.5	54.32% 18.5	16
Multi+$_{Mm}$	57.26% 18.5	10.89% 15.5	46.21% 17.5	77.39% 3.5	54.09% 2.5	41.90% 18.5	77.19% 4.5	54.32% 18.5	16
Multi+$_{MM}$	57.26% 18.5	10.89% 15.5	46.21% 17.5	77.39% 3.5	54.09% 2.5	41.90% 18.5	77.19% 4.5	54.32% 18.5	16
Single±$_{stop}$	74.34% 9.5	17.82% 9.5	53.96% 14.5	46.62% 20.5	47.41% 20.5	65.60% 9.5	46.37% 20.5	54.33% 15.5	19.5
Single±	74.34% 9.5	17.82% 9.5	53.96% 14.5	46.62% 20.5	47.41% 20.5	65.60% 9.5	46.37% 20.5	54.33% 15.5	19.5
Multi±$_{stop,mm}$	74.83% 4.5	18.81% 4.5	58.25% 4.5	48.02% 15.5	49.84% 14.5	65.99% 4.5	48.62% 15.5	55.99% 8.5	4.5
Multi±$_{stop,mM}$	74.83% 4.5	18.81% 4.5	58.25% 4.5	48.02% 15.5	49.84% 14.5	65.99% 4.5	48.62% 15.5	55.99% 8.5	4.5
Multi±$_{stop,Mm}$	74.83% 4.5	18.81% 4.5	58.25% 4.5	48.02% 15.5	49.84% 14.5	65.99% 4.5	48.62% 15.5	55.99% 8.5	4.5
Multi±$_{stop,MM}$	74.83% 4.5	18.81% 4.5	58.25% 4.5	48.02% 15.5	49.84% 14.5	65.99% 4.5	48.62% 15.5	55.99% 8.5	4.5
Multi±$_{mm}$	74.83% 4.5	18.81% 4.5	58.25% 4.5	48.02% 15.5	49.84% 14.5	65.99% 4.5	48.62% 15.5	55.99% 8.5	4.5
Multi±$_{mM}$	74.83% 4.5	18.81% 4.5	58.25% 4.5	48.02% 15.5	49.84% 14.5	65.99% 4.5	48.62% 15.5	55.99% 8.5	4.5
Multi±$_{Mm}$	74.83% 4.5	18.81% 4.5	58.25% 4.5	48.02% 15.5	49.84% 14.5	65.99% 4.5	48.62% 15.5	55.99% 8.5	4.5
Multi±$_{MM}$	74.83% 4.5	18.81% 4.5	58.25% 4.5	48.02% 15.5	49.84% 14.5	65.99% 4.5	48.62% 15.5	55.99% 8.5	4.5
Precision									
BR	69.22% 6	11.88% 6	43.73% 10	47.11% 17	41.84% 17	53.69% 6	47.37% 17	50.33% 11	11
Single+$_{stop}$	62.71% 14	5.45% 10	44.71% 7	58.33% 8	47.53% 13	45.01% 10	59.90% 8	51.40% 10	10
Single+	56.35% 17	4.95% 11	43.72% 11	80.69% 5	53.06% 1	41.71% 11	81.95% 5	55.28% 3	6
Multi+$_{stop,mm}$	66.50% 12.5	10.40% 8	44.50% 8.5	47.52% 15.5	43.52% 15.5	49.13% 8.5	49.62% 15.5	49.38% 16.5	16.5
Multi+$_{stop,mM}$	66.50% 12.5	10.40% 8	44.50% 8.5	47.52% 15.5	43.52% 15.5	49.13% 8.5	49.62% 15.5	49.38% 16.5	16.5
Multi+$_{stop,Mm}$	33.17% 19.5	0.00% 17.5	33.04% 15.5	100.00% 2.5	48.59% 10.5	33.00% 15.5	100.00% 2.5	49.63% 13.5	13.5
Multi+$_{stop,MM}$	33.17% 19.5	0.00% 17.5	33.04% 15.5	100.00% 2.5	48.59% 10.5	33.00% 15.5	100.00% 2.5	49.63% 13.5	13.5
Multi+$_{mm}$	56.77% 15.5	2.97% 12.5	41.74% 12.5	76.32% 6.5	50.45% 5.5	41.63% 12.5	77.94% 6.5	54.28% 7.5	8
Multi+$_{mM}$	56.77% 15.5	2.97% 12.5	41.74% 12.5	76.32% 6.5	50.45% 5.5	41.63% 12.5	77.94% 6.5	54.28% 7.5	8
Multi+$_{Mm}$	33.17% 19.5	0.00% 17.5	33.04% 15.5	100.00% 2.5	48.59% 10.5	33.00% 15.5	100.00% 2.5	49.63% 13.5	13.5
Multi+$_{MM}$	33.17% 19.5	0.00% 17.5	33.04% 15.5	100.00% 2.5	48.59% 10.5	33.00% 15.5	100.00% 2.5	49.63% 13.5	13.5
Single±$_{stop}$	71.53% 3	12.87% 5	54.46% 5	54.29% 9	51.14% 4	57.34% 3	52.88% 11	55.02% 4	3
Single±	69.06% 7	10.40% 8	47.85% 6	49.67% 14	45.72% 14	53.17% 7	50.38% 14	51.74% 9	8
Multi±$_{stop,mm}$	71.53% 3	16.83% 3.5	55.73% 1.5	54.21% 10.5	51.67% 2.5	57.14% 4.5	54.14% 9.5	55.60% 1.5	1.5
Multi±$_{stop,mM}$	71.53% 3	16.83% 3.5	55.73% 1.5	54.21% 10.5	51.67% 2.5	57.14% 4.5	54.14% 9.5	55.60% 1.5	1.5
Multi±$_{stop,Mm}$	67.08% 9.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{stop,MM}$	67.08% 9.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{mm}$	71.53% 3	17.82% 1.5	55.49% 3.5	51.24% 12.5	50.23% 7.5	57.42% 1.5	52.38% 12.5	54.78% 5.5	4.5
Multi±$_{mM}$	71.53% 3	17.82% 1.5	55.49% 3.5	51.24% 12.5	50.23% 7.5	57.42% 1.5	52.38% 12.5	54.78% 5.5	4.5
Multi±$_{Mm}$	67.08% 9.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{MM}$	67.08% 9.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Subset Accuracy									
Multi+$_{stop,Mm}$	72.28% 3	14.85% 3	55.75% 2	58.58% 2	53.64% 1	57.63% 3	59.65% 2	58.62% 1	2
Multi+$_{Mm}$	56.77% 4	11.88% 4	45.30% 4	75.91% 1	53.25% 2	41.45% 4	75.94% 1	53.63% 4	3.5
Multi±$_{stop,Mm}$	75.66% 1	18.81% 1	56.27% 1	49.42% 3	49.83% 3	67.33% 1	50.63% 3	57.80% 2	1
Multi±$_{Mm}$	75.17% 2	17.82% 2	54.25% 3	48.27% 4	48.23% 4	66.55% 2	49.37% 4	56.69% 3	3.5

Table 13: Predictive performance of different multi-label classification approaches on the data set EMOTIONS using the rule-dependent evaluation strategy.

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
F-Measure									
BR	99.91% 1	97.99% 1	99.25% 7	98.91% 6	98.91% 5	99.59% 1	98.37% 4	98.97% 1	1
Single+$_{stop}$	98.46% 13	58.79% 13	80.68% 13	99.58% 1	87.03% 13	75.00% 13	99.18% 1	85.41% 13	10
Single+	99.11% 12	75.88% 12	90.98% 12	97.99% 11	92.98% 12	85.82% 12	96.33% 11	90.77% 12	13
Multi+$_{stop,mm}$	99.85% 6.5	96.98% 4.5	99.25% 7	98.49% 9.5	98.64% 6.5	99.17% 8.5	97.55% 8.5	98.35% 6.5	6.5
Multi+$_{stop,mM}$	99.85% 6.5	96.98% 4.5	99.25% 7	98.49% 9.5	98.64% 6.5	99.17% 8.5	97.55% 8.5	98.35% 6.5	6.5
Multi+$_{stop,Mm}$	22.93% 20.5	0.00% 17.5	5.50% 16.5	99.16% 2.5	10.28% 16.5	5.50% 16.5	98.37% 4	10.43% 16.5	14.5
Multi+$_{stop,MM}$	22.93% 20.5	0.00% 17.5	5.50% 16.5	99.16% 2.5	10.28% 16.5	5.50% 16.5	98.37% 4	10.43% 16.5	14.5
Multi+$_{mm}$	99.26% 10.5	80.40% 10.5	92.21% 10.5	97.57% 12.5	93.36% 10.5	88.97% 10.5	95.51% 12.5	92.13% 10.5	11.5
Multi+$_{mM}$	99.26% 10.5	80.40% 10.5	92.21% 10.5	97.57% 12.5	93.36% 10.5	88.97% 10.5	95.51% 12.5	92.13% 10.5	11.5
Multi+$_{Mm}$	52.08% 18.5	0.00% 17.5	8.18% 14.5	97.19% 14.5	14.86% 14.5	8.18% 14.5	93.06% 14.5	15.04% 14.5	16.5
Multi+$_{MM}$	52.08% 18.5	0.00% 17.5	8.18% 14.5	97.19% 14.5	14.86% 14.5	8.18% 14.5	93.06% 14.5	15.04% 14.5	16.5
Single±$_{stop}$	99.89% 2.5	96.98% 4.5	99.58% 3.5	99.08% 4.5	99.13% 1.5	99.18% 6.5	98.37% 4	98.77% 2.5	2.5
Single±	99.89% 2.5	96.98% 4.5	99.58% 3.5	99.08% 4.5	99.13% 1.5	99.18% 6.5	98.37% 4	98.77% 2.5	2.5
Multi±$_{stop,mm}$	99.87% 4.5	96.98% 4.5	99.75% 1.5	98.83% 7.5	99.05% 3.5	99.58% 2.5	97.55% 8.5	98.56% 4.5	4.5
Multi±$_{stop,mM}$	99.87% 4.5	96.98% 4.5	99.75% 1.5	98.83% 7.5	99.05% 3.5	99.58% 2.5	97.55% 8.5	98.56% 4.5	4.5
Multi±$_{stop,Mm}$	95.44% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{stop,MM}$	95.44% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{mm}$	99.57% 8.5	89.45% 8.5	99.25% 7	95.85% 16.5	97.02% 8.5	99.55% 4.5	91.02% 16.5	95.10% 8.5	8.5
Multi±$_{mM}$	99.57% 8.5	89.45% 8.5	99.25% 7	95.85% 16.5	97.02% 8.5	99.55% 4.5	91.02% 16.5	95.10% 8.5	8.5
Multi±$_{Mm}$	95.44% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{MM}$	95.44% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Hamming Accuracy									
BR	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Single+$_{stop}$	98.40% 20	57.29% 20	80.43% 20	99.33% 1	86.72% 20	74.46% 20	98.78% 1	84.91% 20	16
Single+	94.38% 21	27.14% 21	55.13% 21	98.91% 15	66.30% 21	44.69% 21	97.96% 15	61.38% 21	21
Multi+$_{stop,mm}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi+$_{stop,mM}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi+$_{stop,Mm}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi+$_{stop,MM}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi+$_{mm}$	99.26% 17.5	80.40% 17.5	92.21% 17.5	97.57% 19.5	93.36% 17.5	88.97% 17.5	95.51% 19.5	92.13% 17.5	18.5
Multi+$_{mM}$	99.26% 17.5	80.40% 17.5	92.21% 17.5	97.57% 19.5	93.36% 17.5	88.97% 17.5	95.51% 19.5	92.13% 17.5	18.5
Multi+$_{Mm}$	99.26% 17.5	80.40% 17.5	92.21% 17.5	97.57% 19.5	93.36% 17.5	88.97% 17.5	95.51% 19.5	92.13% 17.5	18.5
Multi+$_{MM}$	99.26% 17.5	80.40% 17.5	92.21% 17.5	97.57% 19.5	93.36% 17.5	88.97% 17.5	95.51% 19.5	92.13% 17.5	18.5
Single±$_{stop}$	99.80% 14.5	94.97% 14.5	98.99% 14.5	97.99% 16.5	98.29% 14.5	99.16% 14.5	96.33% 16.5	97.72% 14.5	14.5
Single±	99.80% 14.5	94.97% 14.5	98.99% 14.5	97.99% 16.5	98.29% 14.5	99.16% 14.5	96.33% 16.5	97.72% 14.5	14.5
Multi±$_{stop,mm}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi±$_{stop,mM}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi±$_{stop,Mm}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi±$_{stop,MM}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi±$_{mm}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi±$_{mM}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi±$_{Mm}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Multi±$_{MM}$	99.91% 7	97.99% 7	99.75% 7	99.16% 8	99.25% 7	99.59% 7	98.37% 8	98.97% 7	7
Precision									
BR	99.70% 8	92.46% 9	98.74% 8	96.78% 13	97.43% 9	99.57% 4	93.88% 10	96.64% 8	9
Single+$_{stop}$	98.03% 13	53.77% 13	78.23% 13	96.19% 14	84.17% 13	71.79% 13	93.47% 11	81.21% 13	13
Single+	99.40% 10	86.93% 10	94.47% 12	95.94% 15	94.23% 10	94.19% 10	92.65% 15	93.42% 10	10
Multi+$_{stop,mm}$	99.81% 2.5	95.48% 4.5	98.74% 8	97.65% 8.5	97.94% 7.5	99.58% 2.5	96.33% 7.5	97.93% 2.5	4.5
Multi+$_{stop,mM}$	99.81% 2.5	95.48% 4.5	98.74% 8	97.65% 8.5	97.94% 7.5	99.58% 2.5	96.33% 7.5	97.93% 2.5	4.5
Multi+$_{stop,Mm}$	22.93% 20.5	0.00% 17.5	5.50% 16.5	99.16% 1.5	10.28% 16.5	5.50% 16.5	98.37% 2	10.43% 16.5	14.5
Multi+$_{stop,MM}$	22.93% 20.5	0.00% 17.5	5.50% 16.5	99.16% 1.5	10.28% 16.5	5.50% 16.5	98.37% 2	10.43% 16.5	14.5
Multi+$_{mm}$	99.11% 11.5	85.93% 11.5	94.64% 10.5	94.10% 16.5	92.68% 11.5	91.56% 11.5	88.57% 16.5	90.04% 11.5	11.5
Multi+$_{mM}$	99.11% 11.5	85.93% 11.5	94.64% 10.5	94.10% 16.5	92.68% 11.5	91.56% 11.5	88.57% 16.5	90.04% 11.5	11.5
Multi+$_{Mm}$	52.08% 18.5	0.00% 17.5	8.18% 14.5	97.19% 11.5	14.86% 14.5	8.18% 14.5	93.06% 13	15.04% 14.5	16.5
Multi+$_{MM}$	52.08% 18.5	0.00% 17.5	8.18% 14.5	97.19% 11.5	14.86% 14.5	8.18% 14.5	93.06% 13	15.04% 14.5	16.5
Single±$_{stop}$	99.85% 1	96.48% 2	98.99% 6	99.08% 3	98.81% 3	98.37% 5	98.37% 2	98.37% 1	1
Single±	99.68% 9	94.47% 8	100.00% 1	97.36% 10	98.22% 6	100.00% 1	93.06% 13	96.41% 9	8
Multi±$_{stop,mm}$	99.80% 4.5	96.48% 2	99.16% 4.5	98.99% 4.5	98.86% 1.5	97.56% 8.5	97.96% 4.5	97.76% 4.5	2.5
Multi±$_{stop,mM}$	99.80% 4.5	96.48% 2	99.16% 4.5	98.99% 4.5	98.86% 1.5	97.56% 8.5	97.96% 4.5	97.76% 4.5	2.5
Multi±$_{stop,Mm}$	95.44% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{stop,MM}$	95.44% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{mm}$	99.76% 6.5	94.97% 6.5	99.45% 2.5	98.49% 6.5	98.77% 4.5	98.33% 6.5	96.33% 7.5	97.32% 6.5	6.5
Multi±$_{mM}$	99.76% 6.5	94.97% 6.5	99.45% 2.5	98.49% 6.5	98.77% 4.5	98.33% 6.5	96.33% 7.5	97.32% 6.5	6.5
Multi±$_{Mm}$	95.44% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{MM}$	95.44% 15.5	0.00% 17.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Subset Accuracy									
Multi+$_{stop,Mm}$	99.91% 2	97.99% 2	99.75% 2	99.16% 2	99.25% 2	99.59% 2	98.37% 2	98.97% 2	2
Multi+$_{Mm}$	97.95% 4	46.23% 4	75.82% 4	97.74% 4	82.84% 4	70.15% 4	95.92% 4	81.03% 4	4
Multi±$_{stop,Mm}$	99.91% 2	97.99% 2	99.75% 2	99.16% 2	99.25% 2	99.59% 2	98.37% 2	98.97% 2	2
Multi±$_{Mm}$	99.91% 2	97.99% 2	99.75% 2	99.16% 2	99.25% 2	99.59% 2	98.37% 2	98.97% 2	2

Table 14: Predictive performance of different multi-label classification approaches on the data set GENBASE using the rule-dependent evaluation strategy.

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
F-Measure									
BR	93.91% 7	41.49% 9	56.87% 5	56.04% 9	55.22% 6	40.56% 5	41.85% 10	41.85% 8	4
Single+$_{stop}$	89.93% 14	2.48% 16	14.27% 16	19.39% 21	15.56% 16	22.12% 12	38.66% 17	38.66% 15	19
Single+	84.41% 17	0.31% 18	9.43% 17	23.47% 20	12.73% 17	15.96% 15	48.24% 7	48.24% 5	18
Multi+$_{stop,mm}$	93.48% 12.5	39.94% 11.5	54.23% 8.5	55.74% 10.5	53.49% 8.5	37.24% 10.5	40.58% 13.5	40.58% 11.5	12.5
Multi+$_{stop,mM}$	93.48% 12.5	39.94% 11.5	54.23% 8.5	55.74% 10.5	53.49% 8.5	37.24% 10.5	40.58% 13.5	40.58% 11.5	12.5
Multi+$_{stop,Mm}$	5.39% 20.5	0.31% 18	5.39% 20.5	53.25% 12.5	9.41% 20.5	5.12% 18.5	100.00% 1.5	9.37% 16.5	20.5
Multi+$_{stop,MM}$	5.39% 20.5	0.31% 18	5.39% 20.5	53.25% 12.5	9.41% 20.5	5.12% 18.5	100.00% 1.5	9.37% 16.5	20.5
Multi+$_{mm}$	86.18% 15.5	32.51% 14.5	42.34% 14.5	60.53% 1.5	45.92% 14.5	20.51% 13.5	59.42% 5.5	59.42% 3.5	10.5
Multi+$_{mM}$	86.18% 15.5	32.51% 14.5	42.34% 14.5	60.53% 1.5	45.92% 14.5	20.51% 13.5	59.42% 5.5	59.42% 3.5	10.5
Multi+$_{Mm}$	32.69% 18.5	0.00% 20.5	6.72% 18.5	49.07% 14.5	11.49% 18.5	6.63% 16.5	93.29% 3.5	93.29% 1.5	15.5
Multi+$_{MM}$	32.69% 18.5	0.00% 20.5	6.72% 18.5	49.07% 14.5	11.49% 18.5	6.63% 16.5	93.29% 3.5	93.29% 1.5	15.5
Single±$_{stop}$	94.02% 6	39.32% 13	55.46% 7	56.79% 8	54.57% 7	41.35% 4	41.21% 11.5	41.21% 9.5	5
Single±	94.10% 5	41.18% 10	56.79% 6	57.56% 7	55.65% 5	42.02% 3	41.21% 11.5	41.21% 9.5	3
Multi±$_{stop,mm}$	93.87% 8.5	43.03% 6.5	59.58% 1.5	59.33% 3.5	57.90% 1.5	40.54% 6.5	43.13% 8.5	43.13% 6.5	1.5
Multi±$_{stop,mM}$	93.87% 8.5	43.03% 6.5	59.58% 1.5	59.33% 3.5	57.90% 1.5	40.54% 6.5	43.13% 8.5	43.13% 6.5	1.5
Multi±$_{stop,Mm}$	94.92% 1.5	47.06% 1.5	47.37% 10.5	47.21% 16.5	47.27% 10.5	66.67% 1.5	0.64% 18.5	1.27% 18.5	9
Multi±$_{stop,MM}$	94.92% 1.5	47.06% 1.5	47.37% 10.5	47.21% 16.5	47.27% 10.5	66.67% 1.5	0.64% 18.5	1.27% 18.5	8
Multi±$_{mm}$	93.74% 10.5	43.03% 6.5	57.60% 3.5	58.11% 5.5	56.44% 3.5	38.73% 8.5	38.98% 15.5	38.98% 13.5	6.5
Multi±$_{mM}$	93.74% 10.5	43.03% 6.5	57.60% 3.5	58.11% 5.5	56.44% 3.5	38.73% 8.5	38.98% 15.5	38.98% 13.5	6.5
Multi±$_{Mm}$	94.90% 3.5	46.75% 3.5	46.75% 12.5	46.75% 18.5	46.75% 12.5	0.00% 20.5	0.00% 20.5	0.00% 20.5	15.5
Multi±$_{MM}$	94.90% 3.5	46.75% 3.5	46.75% 12.5	46.75% 18.5	46.75% 12.5	0.00% 20.5	0.00% 20.5	0.00% 20.5	15.5
Hamming Accuracy									
BR	93.16% 15	39.01% 11	52.59% 11	53.51% 15	51.66% 11	51.66% 11	39.62% 7	37.13% 5	11
Single+$_{stop}$	67.05% 21	1.55% 20	12.41% 20	28.78% 20	15.06% 20	15.06% 20	56.23% 1	14.83% 21	20
Single+	85.25% 20	0.00% 21	10.84% 21	25.03% 21	14.42% 21	14.42% 21	47.28% 6	24.65% 20	21
Multi+$_{stop,mm}$	93.29% 12.5	37.46% 13.5	51.68% 13.5	51.34% 17.5	50.16% 13.5	50.16% 13.5	35.14% 9.5	34.81% 17.5	17.5
Multi+$_{stop,mM}$	93.29% 12.5	37.46% 13.5	51.68% 13.5	51.34% 17.5	50.16% 13.5	50.16% 13.5	35.14% 9.5	34.81% 17.5	17.5
Multi+$_{stop,Mm}$	93.29% 12.5	37.46% 13.5	51.68% 13.5	51.34% 17.5	50.16% 13.5	50.16% 13.5	35.14% 9.5	34.81% 17.5	17.5
Multi+$_{stop,MM}$	93.29% 12.5	37.46% 13.5	51.68% 13.5	51.34% 17.5	50.16% 13.5	50.16% 13.5	35.14% 9.5	34.81% 17.5	17.5
Multi+$_{mm}$	92.10% 17.5	34.06% 17.5	49.35% 17.5	55.00% 12.5	50.06% 17.5	50.06% 17.5	47.92% 3.5	38.22% 2.5	13.5
Multi+$_{mM}$	92.10% 17.5	34.06% 17.5	49.35% 17.5	55.00% 12.5	50.06% 17.5	50.06% 17.5	47.92% 3.5	38.22% 2.5	13.5
Multi+$_{Mm}$	92.10% 17.5	34.06% 17.5	49.35% 17.5	55.00% 12.5	50.06% 17.5	50.06% 17.5	47.92% 3.5	38.22% 2.5	13.5
Multi+$_{MM}$	92.10% 17.5	34.06% 17.5	49.35% 17.5	55.00% 12.5	50.06% 17.5	50.06% 17.5	47.92% 3.5	38.22% 2.5	13.5
Single±$_{stop}$	94.25% 1.5	46.44% 1.5	59.83% 1.5	56.37% 1.5	56.72% 1.5	56.72% 1.5	31.95% 20.5	36.17% 14.5	1.5
Single±	94.25% 1.5	46.44% 1.5	59.83% 1.5	56.37% 1.5	56.72% 1.5	56.72% 1.5	31.95% 20.5	36.17% 14.5	1.5
Multi±$_{stop,mm}$	94.18% 6.5	44.27% 6.5	58.46% 6.5	55.52% 6.5	55.58% 6.5	55.58% 6.5	32.91% 15.5	36.59% 9.5	6.5
Multi±$_{stop,mM}$	94.18% 6.5	44.27% 6.5	58.46% 6.5	55.52% 6.5	55.58% 6.5	55.58% 6.5	32.91% 15.5	36.59% 9.5	6.5
Multi±$_{stop,Mm}$	94.18% 6.5	44.27% 6.5	58.46% 6.5	55.52% 6.5	55.58% 6.5	55.58% 6.5	32.91% 15.5	36.59% 9.5	6.5
Multi±$_{stop,MM}$	94.18% 6.5	44.27% 6.5	58.46% 6.5	55.52% 6.5	55.58% 6.5	55.58% 6.5	32.91% 15.5	36.59% 9.5	6.5
Multi±$_{mm}$	94.18% 6.5	44.27% 6.5	58.46% 6.5	55.52% 6.5	55.58% 6.5	55.58% 6.5	32.91% 15.5	36.59% 9.5	6.5
Multi±$_{mM}$	94.18% 6.5	44.27% 6.5	58.46% 6.5	55.52% 6.5	55.58% 6.5	55.58% 6.5	32.91% 15.5	36.59% 9.5	6.5
Multi±$_{Mm}$	94.18% 6.5	44.27% 6.5	58.46% 6.5	55.52% 6.5	55.58% 6.5	55.58% 6.5	32.91% 15.5	36.59% 9.5	6.5
Multi±$_{MM}$	94.18% 6.5	44.27% 6.5	58.46% 6.5	55.52% 6.5	55.58% 6.5	55.58% 6.5	32.91% 15.5	36.59% 9.5	6.5
Precision									
BR	92.50% 11	38.70% 11	51.02% 7	52.87% 11	50.35% 7	30.81% 9	37.70% 8.5	37.70% 8.5	7
Single+$_{stop}$	89.23% 14	2.48% 16	14.83% 16	20.06% 21	15.88% 16	20.21% 12	37.70% 8.5	37.70% 8.5	17
Single+	74.71% 17	0.31% 18	7.20% 17	31.42% 20	11.08% 19	11.97% 15	62.30% 5	62.30% 5	21
Multi+$_{stop,mm}$	92.29% 12.5	34.67% 12.5	48.53% 8.5	50.17% 12.5	47.68% 8.5	28.49% 10.5	33.87% 10.5	33.87% 10.5	10.5
Multi+$_{stop,mM}$	92.29% 12.5	34.67% 12.5	48.53% 8.5	50.17% 12.5	47.68% 8.5	28.49% 10.5	33.87% 10.5	33.87% 10.5	10.5
Multi+$_{stop,Mm}$	5.39% 20.5	0.31% 18	5.39% 20.5	53.25% 9.5	9.41% 20.5	5.12% 18.5	100.00% 1.5	100.00% 1.5	14.5
Multi+$_{stop,MM}$	5.39% 20.5	0.31% 18	5.39% 20.5	53.25% 9.5	9.41% 20.5	5.12% 18.5	100.00% 1.5	100.00% 1.5	14.5
Multi+$_{mm}$	85.92% 15.5	33.75% 14.5	42.78% 14.5	57.08% 6.5	45.63% 14.5	17.63% 13.5	47.92% 6.5	47.92% 6.5	12.5
Multi+$_{mM}$	85.92% 15.5	33.75% 14.5	42.78% 14.5	57.08% 6.5	45.63% 14.5	17.63% 13.5	47.92% 6.5	47.92% 6.5	12.5
Multi+$_{Mm}$	32.69% 18.5	0.00% 20.5	6.72% 18.5	49.07% 14.5	11.49% 17.5	6.63% 16.5	93.29% 3.5	93.29% 3.5	19.5
Multi+$_{MM}$	32.69% 18.5	0.00% 20.5	6.72% 18.5	49.07% 14.5	11.49% 17.5	6.63% 16.5	93.29% 3.5	93.29% 3.5	19.5
Single±$_{stop}$	94.15% 6	45.51% 8.5	58.23% 6	56.68% 8	56.26% 6	40.80% 4	32.59% 17	32.59% 17	6
Single±	94.43% 5	45.51% 8.5	59.29% 1	57.62% 5	57.12% 3	43.93% 3	33.55% 13	33.55% 13	1
Multi±$_{stop,mm}$	93.32% 9.5	45.51% 8.5	59.28% 2.5	58.47% 3.5	56.93% 4.5	34.10% 7.5	33.23% 15.5	33.23% 15.5	4.5
Multi±$_{stop,mM}$	93.32% 9.5	45.51% 8.5	59.28% 2.5	58.47% 3.5	56.93% 4.5	34.10% 7.5	33.23% 15.5	33.23% 15.5	4.5
Multi±$_{stop,Mm}$	94.92% 1.5	47.06% 1.5	47.37% 10.5	47.21% 16.5	47.27% 10.5	66.67% 1.5	0.64% 18.5	0.64% 18.5	8.5
Multi±$_{stop,MM}$	94.92% 1.5	47.06% 1.5	47.37% 10.5	47.21% 16.5	47.27% 10.5	66.67% 1.5	0.64% 18.5	0.64% 18.5	8.5
Multi±$_{mm}$	93.79% 7.5	46.44% 5.5	58.24% 4.5	59.34% 1.5	57.21% 1.5	37.77% 5.5	33.55% 13	33.55% 13	2.5
Multi±$_{mM}$	93.79% 7.5	46.44% 5.5	58.24% 4.5	59.34% 1.5	57.21% 1.5	37.77% 5.5	33.55% 13	33.55% 13	2.5
Multi±$_{Mm}$	94.90% 3.5	46.75% 3.5	46.75% 12.5	46.75% 18.5	46.75% 12.5	0.00% 20.5	0.00% 20.5	0.00% 20.5	17
Multi±$_{MM}$	94.90% 3.5	46.75% 3.5	46.75% 12.5	46.75% 18.5	46.75% 12.5	0.00% 20.5	0.00% 20.5	0.00% 20.5	17
Subset Accuracy									
Multi+$_{stop,Mm}$	93.30% 3	38.39% 3	52.51% 3	53.13% 3	51.46% 3	35.24% 3	37.38% 2	36.28% 4	3
Multi+$_{Mm}$	91.49% 4	30.65% 4	46.03% 4	52.42% 4	46.93% 4	29.63% 4	48.56% 1	36.80% 1	4
Multi±$_{stop,Mm}$	94.18% 1.5	44.27% 1.5	58.46% 1.5	55.52% 1.5	55.58% 1.5	41.20% 1.5	32.91% 3.5	36.59% 2.5	1.5
Multi±$_{Mm}$	94.18% 1.5	44.27% 1.5	58.46% 1.5	55.52% 1.5	55.58% 1.5	41.20% 1.5	32.91% 3.5	36.59% 2.5	1.5

Table 15: Predictive performance of different multi-label classification approaches on the data set BIRDS using the rule-dependent evaluation strategy.

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
F-Measure									
BR	83.04% 8	83.04% 5	45.41% 9	45.41% 13	47.97% 12	53.06% 8	53.06% 11	53.92% 9	9
Single+$_{stop}$	78.55% 14	78.55% 9	44.38% 10	44.38% 14	50.02% 10	43.46% 10	43.46% 14	50.91% 12	10
Single+	80.39% 13	80.39% 8	47.13% 8	47.13% 12	53.76% 3	47.16% 9	47.16% 13	56.02% 8	8
Multi+$_{stop,mm}$	84.11% 6.5	84.11% 3.5	51.24% 4.5	51.24% 9.5	53.93% 1.5	55.72% 6.5	55.72% 7.5	57.62% 5.5	5.5
Multi+$_{stop,mM}$	84.11% 6.5	84.11% 3.5	51.24% 4.5	51.24% 9.5	53.93% 1.5	55.72% 6.5	55.72% 7.5	57.62% 5.5	5.5
Multi+$_{stop,Mm}$	18.53% 19.5	0.17% 16.5	18.14% 15.5	99.41% 1.5	30.37% 15.5	18.12% 15.5	99.46% 1.5	30.65% 15.5	15.5
Multi+$_{stop,MM}$	18.53% 19.5	0.17% 16.5	18.14% 15.5	99.41% 1.5	30.37% 15.5	18.12% 15.5	99.46% 1.5	30.65% 15.5	15.5
Multi+$_{mm}$	76.84% 15.5	76.84% 10.5	44.19% 11.5	44.19% 15.5	52.46% 6.5	42.04% 11.5	42.04% 15.5	53.55% 10.5	13.5
Multi+$_{mM}$	76.84% 15.5	76.84% 10.5	44.19% 11.5	44.19% 15.5	52.46% 6.5	42.04% 11.5	42.04% 15.5	53.55% 10.5	13.5
Multi+$_{Mm}$	19.52% 17.5	0.17% 16.5	18.33% 13.5	99.08% 3.5	30.60% 13.5	18.26% 13.5	99.08% 3.5	30.83% 13.5	11.5
Multi+$_{MM}$	19.52% 17.5	0.17% 16.5	18.33% 13.5	99.08% 3.5	30.60% 13.5	18.26% 13.5	99.08% 3.5	30.83% 13.5	11.5
Single±$_{stop}$	86.55% 2	86.55% 2	51.94% 3	51.94% 8	52.13% 8	65.90% 2	65.90% 6	58.92% 1	1
Single±	86.72% 1	86.72% 1	50.71% 6	50.71% 11	50.67% 9	67.47% 1	67.47% 5	58.37% 4	2
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	85.49% 5	40.22% 14	48.86% 7	52.26% 7	49.36% 11	62.03% 5	51.19% 12	56.09% 7	7
Multi±$_{stop,Mm}$	81.90% 10.5	0.00% 19.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	19.5
Multi±$_{stop,MM}$	81.90% 10.5	81.90% 6.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	17.5
Multi±$_{mm}$	86.54% 3.5	46.07% 12.5	53.41% 1.5	54.47% 5.5	53.11% 4.5	65.87% 3.5	53.19% 9.5	58.86% 2.5	3.5
Multi±$_{mM}$	86.54% 3.5	46.07% 12.5	53.41% 1.5	54.47% 5.5	53.11% 4.5	65.87% 3.5	53.19% 9.5	58.86% 2.5	3.5
Multi±$_{Mm}$	81.90% 10.5	0.00% 19.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	19.5
Multi±$_{MM}$	81.90% 10.5	81.90% 6.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	0.00% 18.5	17.5
Hamming Accuracy									
BR	78.47% 15	20.65% 15	20.65% 15	51.38% 11	39.33% 11	42.13% 15	50.65% 11	46.00% 12	20
Single+$_{stop}$	66.92% 21	18.90% 16	18.90% 16	71.66% 6	45.46% 10	31.81% 21	72.36% 6	44.19% 13	21
Single+	70.57% 16	14.13% 17	14.13% 17	73.41% 5	47.29% 5	34.93% 16	72.52% 5	47.15% 5	7
Multi+$_{stop,mm}$	81.56% 12.5	34.36% 2.5	34.36% 4.5	56.06% 8.5	49.62% 2.5	49.18% 12.5	55.35% 8.5	52.08% 2.5	2.5
Multi+$_{stop,mM}$	81.56% 12.5	34.36% 2.5	34.36% 4.5	56.06% 8.5	49.62% 2.5	49.18% 12.5	55.35% 8.5	52.08% 2.5	2.5
Multi+$_{stop,Mm}$	81.56% 12.5	34.36% 2.5	34.36% 4.5	56.06% 8.5	49.62% 2.5	49.18% 12.5	55.35% 8.5	52.08% 2.5	2.5
Multi+$_{stop,MM}$	81.56% 12.5	34.36% 2.5	34.36% 4.5	56.06% 8.5	49.62% 2.5	49.18% 12.5	55.35% 8.5	52.08% 2.5	2.5
Multi+$_{mm}$	69.50% 18.5	8.78% 19.5	8.78% 19.5	74.50% 2.5	46.54% 7.5	34.24% 18.5	74.44% 2.5	46.91% 7.5	9.5
Multi+$_{mM}$	69.50% 18.5	8.78% 19.5	8.78% 19.5	74.50% 2.5	46.54% 7.5	34.24% 18.5	74.44% 2.5	46.91% 7.5	9.5
Multi+$_{Mm}$	69.50% 18.5	8.78% 19.5	8.78% 19.5	74.50% 2.5	46.54% 7.5	34.24% 18.5	74.44% 2.5	46.91% 7.5	9.5
Multi+$_{MM}$	69.50% 18.5	8.78% 19.5	8.78% 19.5	74.50% 2.5	46.54% 7.5	34.24% 18.5	74.44% 2.5	46.91% 7.5	9.5
Single±$_{stop}$	85.40% 1.5	28.18% 5.5	34.56% 1.5	35.20% 12.5	34.14% 12.5	69.10% 1.5	34.95% 12.5	46.42% 10.5	5.5
Single±	85.40% 1.5	28.18% 5.5	34.56% 1.5	35.20% 12.5	34.14% 12.5	69.10% 1.5	34.95% 12.5	46.42% 10.5	5.5
Multi±$_{stop,mm}$	84.70% 6.5	24.58% 10.5	24.58% 10.5	30.89% 17.5	29.82% 17.5	66.95% 6.5	30.56% 17.5	41.97% 17.5	15.5
Multi±$_{stop,mM}$	84.70% 6.5	24.58% 10.5	24.58% 10.5	30.89% 17.5	29.82% 17.5	66.95% 6.5	30.56% 17.5	41.97% 17.5	15.5
Multi±$_{stop,Mm}$	84.70% 6.5	24.58% 10.5	24.58% 10.5	30.89% 17.5	29.82% 17.5	66.95% 6.5	30.56% 17.5	41.97% 17.5	15.5
Multi±$_{stop,MM}$	84.70% 6.5	24.58% 10.5	24.58% 10.5	30.89% 17.5	29.82% 17.5	66.95% 6.5	30.56% 17.5	41.97% 17.5	15.5
Multi±$_{mm}$	84.70% 6.5	24.58% 10.5	24.58% 10.5	30.89% 17.5	29.82% 17.5	66.95% 6.5	30.56% 17.5	41.97% 17.5	15.5
Multi±$_{mM}$	84.70% 6.5	24.58% 10.5	24.58% 10.5	30.89% 17.5	29.82% 17.5	66.95% 6.5	30.56% 17.5	41.97% 17.5	15.5
Multi±$_{Mm}$	84.70% 6.5	24.58% 10.5	24.58% 10.5	30.89% 17.5	29.82% 17.5	66.95% 6.5	30.56% 17.5	41.97% 17.5	15.5
Multi±$_{MM}$	84.70% 6.5	24.58% 10.5	24.58% 10.5	30.89% 17.5	29.82% 17.5	66.95% 6.5	30.56% 17.5	41.97% 17.5	15.5
Precision									
BR	78.09% 13	78.09% 9	35.56% 10	51.63% 14	39.63% 10	41.34% 9	41.34% 9	45.34% 7	9
Single+$_{stop}$	77.63% 14	77.63% 10	40.06% 9	47.70% 15	41.98% 7	40.01% 10	40.01% 10	43.31% 10	10
Single+	51.90% 15	51.90% 11	25.44% 13	79.35% 5	36.14% 13	24.36% 11	24.36% 11	37.21% 11	11
Multi+$_{stop,mm}$	78.58% 11.5	78.58% 7.5	40.41% 7.5	46.24% 16.5	41.78% 8.5	41.64% 7.5	41.64% 7.5	43.55% 8.5	7.5
Multi+$_{stop,mM}$	78.58% 11.5	78.58% 7.5	40.41% 7.5	46.24% 16.5	41.78% 8.5	41.64% 7.5	41.64% 7.5	43.55% 8.5	7.5
Multi+$_{stop,Mm}$	18.53% 20.5	18.53% 20.5	18.14% 16.5	99.41% 1.5	30.37% 16.5	18.12% 16.5	18.12% 16.5	30.65% 16.5	16.5
Multi+$_{stop,MM}$	18.53% 20.5	18.53% 20.5	18.14% 16.5	99.41% 1.5	30.37% 16.5	18.12% 16.5	18.12% 16.5	30.65% 16.5	16.5
Multi+$_{mm}$	51.38% 16.5	51.38% 12.5	25.89% 11.5	78.22% 6.5	36.24% 11.5	23.95% 12.5	23.95% 12.5	36.60% 12.5	12.5
Multi+$_{mM}$	51.38% 16.5	51.38% 12.5	25.89% 11.5	78.22% 6.5	36.24% 11.5	23.95% 12.5	23.95% 12.5	36.60% 12.5	12.5
Multi+$_{Mm}$	19.52% 18.5	19.52% 18.5	18.33% 14.5	99.08% 3.5	30.60% 14.5	18.26% 14.5	18.26% 14.5	30.83% 14.5	14.5
Multi+$_{MM}$	19.52% 18.5	19.52% 18.5	18.33% 14.5	99.08% 3.5	30.60% 14.5	18.26% 14.5	18.26% 14.5	30.83% 14.5	14.5
Single±$_{stop}$	85.56% 6	85.56% 2	51.73% 6	53.97% 11	51.74% 6	61.81% 6	61.81% 2	57.05% 6	4
Single±	85.76% 5	85.76% 1	52.31% 5	54.77% 10	52.37% 3	62.42% 5	62.42% 1	57.66% 5	3
Multi±$_{stop,mm}$	86.40% 2	47.32% 14.5	54.58% 1.5	55.73% 8.5	54.28% 1.5	64.75% 1.5	54.58% 3	59.23% 1.5	1.5
Multi±$_{stop,mM}$	86.64% 1	47.32% 14.5	54.58% 1.5	55.73% 8.5	54.28% 1.5	64.75% 1.5	54.58% 4	59.23% 1.5	1.5
Multi±$_{stop,Mm}$	81.90% 8.5	81.90% 4.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{stop,MM}$	81.90% 8.5	81.90% 4.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{mm}$	86.13% 3.5	45.57% 16.5	52.95% 3.5	53.34% 12.5	52.32% 4.5	64.45% 3.5	52.19% 5.5	57.68% 3.5	5.5
Multi±$_{mM}$	86.13% 3.5	45.57% 16.5	52.95% 3.5	53.34% 12.5	52.32% 4.5	64.45% 3.5	52.19% 5.5	57.68% 3.5	5.5
Multi±$_{Mm}$	81.90% 8.5	81.90% 4.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Multi±$_{MM}$	81.90% 8.5	81.90% 4.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	0.00% 19.5	19.5
Subset Accuracy									
Multi+$_{stop,Mm}$	81.76% 3	34.95% 1	47.09% 1	55.73% 2	49.34% 1	49.65% 3	54.73% 2	52.07% 1	1
Multi+$_{Mm}$	69.16% 4	8.95% 4	36.14% 2	74.46% 1	46.37% 2	33.95% 4	74.44% 1	46.64% 2	2
Multi±$_{stop,Mm}$	84.77% 1.5	24.75% 2.5	30.22% 3.5	30.98% 3.5	29.90% 3.5	67.40% 1.5	30.72% 3.5	42.20% 3.5	3.5
Multi±$_{Mm}$	84.77% 1.5	24.75% 2.5	30.22% 3.5	30.98% 3.5	29.90% 3.5	67.40% 1.5	30.72% 3.5	42.20% 3.5	3.5

Table 16: Predictive performance of different multi-label classification approaches on the data set SCENE using the rule-dependent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
	F-Measure								
Multi+$_{stop,mm}$	4.78% 8	0.00% 4.5	2.86% 8	99.84% 1	5.53% 8	2.81% 8	99.88% 1	5.47% 8	8
Multi+$_{stop,mM}$	28.78% 5	0.00% 4.5	3.71% 5	97.91% 4	7.11% 5	3.64% 5	97.50% 4	7.02% 5	5
Multi+$_{stop,Mm}$	25.02% 6	0.00% 4.5	3.49% 6	98.76% 2	6.71% 6	3.50% 6	98.50% 2	6.75% 6	6
Multi+$_{stop,MM}$	25.00% 7	0.00% 4.5	3.48% 7	98.53% 3	6.69% 7	3.48% 7	98.13% 3	6.73% 7	7
Multi+$_{mm}$	64.78% 2	0.00% 4.5	8.76% 2	90.70% 8	14.82% 2	6.57% 2	89.13% 8	12.24% 2	3
Multi+$_{mM}$	69.86% 1	0.00% 4.5	9.91% 1	91.63% 7	16.83% 1	7.75% 1	91.13% 7	14.29% 1	1
Multi+$_{Mm}$	58.08% 3	0.00% 4.5	5.81% 3	93.62% 5	10.87% 3	5.81% 3	93.50% 5	10.95% 3	2
Multi+$_{MM}$	58.00% 4	0.00% 4.5	5.75% 4	93.07% 6	10.76% 4	5.76% 4	92.63% 6	10.84% 4	4
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
	Hamming Accuracy								
Multi+$_{stop,mm}$	96.39% 6.5	12.56% 6.5	46.06% 6.5	82.12% 6.5	56.69% 6.5	41.99% 6.5	81.63% 6.5	55.46% 6.5	6.5
Multi+$_{stop,mM}$	96.39% 6.5	12.56% 6.5	46.06% 6.5	82.12% 6.5	56.69% 6.5	41.99% 6.5	81.63% 6.5	55.46% 6.5	6.5
Multi+$_{stop,Mm}$	96.39% 6.5	12.56% 6.5	46.06% 6.5	82.12% 6.5	56.69% 6.5	41.99% 6.5	81.63% 6.5	55.46% 6.5	6.5
Multi+$_{stop,MM}$	96.39% 6.5	12.56% 6.5	46.06% 6.5	82.12% 6.5	56.69% 6.5	41.99% 6.5	81.63% 6.5	55.46% 6.5	6.5
Multi+$_{mm}$	97.01% 2.5	25.58% 2.5	53.96% 2.5	88.24% 2.5	63.28% 2.5	47.68% 2.5	87.50% 2.5	61.73% 2.5	2.5
Multi+$_{mM}$	97.01% 2.5	25.58% 2.5	53.96% 2.5	88.24% 2.5	63.28% 2.5	47.68% 2.5	87.50% 2.5	61.73% 2.5	2.5
Multi+$_{Mm}$	97.01% 2.5	25.58% 2.5	53.96% 2.5	88.24% 2.5	63.28% 2.5	47.68% 2.5	87.50% 2.5	61.73% 2.5	2.5
Multi+$_{MM}$	97.01% 2.5	25.58% 2.5	53.96% 2.5	88.24% 2.5	63.28% 2.5	47.68% 2.5	87.50% 2.5	61.73% 2.5	2.5
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
	Precision								
Multi+$_{stop,mm}$	96.90% 1.5	26.36% 1.5	38.06% 1.5	36.51% 7.5	36.07% 1.5	42.33% 1.5	34.50% 7.5	38.02% 1.5	1.5
Multi+$_{stop,mM}$	96.90% 1.5	26.36% 1.5	38.06% 1.5	36.51% 7.5	36.07% 1.5	42.33% 1.5	34.50% 7.5	38.02% 1.5	1.5
Multi+$_{stop,Mm}$	25.00% 7.5	0.00% 6.5	3.48% 7.5	98.53% 1.5	6.69% 7.5	3.48% 8	98.13% 1.5	6.73% 7.5	7.5
Multi+$_{stop,MM}$	25.00% 7.5	0.00% 6.5	3.48% 7.5	98.53% 1.5	6.69% 7.5	3.48% 8	98.13% 1.5	6.73% 7.5	7.5
Multi+$_{mm}$	94.20% 3.5	9.15% 3.5	28.51% 3.5	60.65% 5.5	35.29% 3.5	25.94% 3.5	59.50% 5.5	36.13% 3.5	3.5
Multi+$_{mM}$	94.20% 3.5	9.15% 3.5	28.51% 3.5	60.65% 5.5	35.29% 3.5	25.94% 3.5	59.50% 5.5	36.13% 3.5	3.5
Multi+$_{Mm}$	58.00% 5.5	0.00% 6.5	5.75% 5.5	93.07% 3.5	10.76% 5.5	5.76% 8	92.63% 3.5	10.84% 5.5	6
Multi+$_{MM}$	58.00% 5.5	0.00% 6.5	5.75% 5.5	93.07% 3.5	10.76% 5.5	5.76% 5	92.63% 3.5	10.84% 5.5	5
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
	Subset Accuracy								
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	

Table 17: Predictive performance of different multi-label classification approaches on the data set MEDICAL using the rule-independent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
	F-Measure								
Multi+$_{stop,mm}$	34.90% 11	0.00% 11.5	33.61% 3	99.34% 6	49.10% 3	33.53% 7	99.50% 6	50.16% 3	3
Multi+$_{stop,mM}$	32.92% 14	0.00% 11.5	32.92% 6	100.00% 3	48.48% 6	32.92% 10	100.00% 3	49.53% 6	8
Multi+$_{stop,Mm}$	32.92% 14	0.00% 11.5	32.92% 6	100.00% 3	48.48% 6	32.92% 10	100.00% 3	49.53% 6	8
Multi+$_{stop,MM}$	32.92% 14	0.00% 11.5	32.92% 6	100.00% 3	48.48% 6	32.92% 10	100.00% 3	49.53% 6	8
Multi+$_{mm}$	52.97% 10	1.49% 5.5	41.43% 2	91.34% 8	55.21% 2	40.55% 6	91.98% 8	56.29% 2	2
Multi+$_{mM}$	54.37% 9	1.49% 5.5	42.82% 1	92.00% 7	56.40% 1	41.37% 5	92.48% 7	57.16% 1	1
Multi+$_{Mm}$	32.92% 14	0.00% 11.5	32.92% 6	100.00% 3	48.48% 6	32.92% 10	100.00% 3	49.53% 6	8
Multi+$_{MM}$	32.92% 14	0.00% 11.5	32.92% 6	100.00% 3	48.48% 6	32.92% 10	100.00% 3	49.53% 6	8
Multi±$_{stop,mm}$	72.03% 1.5	8.42% 1.5	13.86% 9.5	17.33% 9.5	15.10% 9.5	77.78% 3.5	21.05% 9.5	33.14% 9.5	4.5
Multi±$_{stop,mM}$	70.71% 3.5	5.45% 3.5	9.08% 11.5	10.89% 11.5	9.80% 11.5	83.33% 1.5	13.78% 11.5	23.66% 11.5	11.5
Multi±$_{stop,Mm}$	67.08% 6.5	0.00% 11.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	14.5
Multi±$_{stop,MM}$	67.08% 6.5	0.00% 11.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	14.5
Multi±$_{mm}$	72.03% 1.5	8.42% 1.5	13.86% 9.5	17.33% 9.5	15.10% 9.5	77.78% 3.5	21.05% 9.5	33.14% 9.5	4.5
Multi±$_{mM}$	70.71% 3.5	5.45% 3.5	9.08% 11.5	10.89% 11.5	9.80% 11.5	83.33% 1.5	13.78% 11.5	23.66% 11.5	11.5
Multi±$_{Mm}$	67.08% 6.5	0.00% 11.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	14.5
Multi±$_{MM}$	67.08% 6.5	0.00% 11.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	14.5
	Hamming Accuracy								
Multi+$_{stop,mm}$	44.97% 14.5	0.99% 12.5	37.54% 14.5	91.91% 6.5	51.17% 14.5	36.63% 14.5	91.98% 6.5	52.39% 14.5	14.5
Multi+$_{stop,mM}$	44.97% 14.5	0.99% 12.5	37.54% 14.5	91.91% 6.5	51.17% 14.5	36.63% 14.5	91.98% 6.5	52.39% 14.5	14.5
Multi+$_{stop,Mm}$	44.97% 14.5	0.99% 12.5	37.54% 14.5	91.91% 6.5	51.17% 14.5	36.63% 14.5	91.98% 6.5	52.39% 14.5	14.5
Multi+$_{stop,MM}$	44.97% 14.5	0.99% 12.5	37.54% 14.5	91.91% 6.5	51.17% 14.5	36.63% 14.5	91.98% 6.5	52.39% 14.5	14.5
Multi+$_{mm}$	56.19% 10.5	0.99% 12.5	43.00% 10.5	92.00% 2.5	56.98% 2.5	42.45% 10.5	92.98% 2.5	58.29% 2.5	2.5
Multi+$_{mM}$	56.19% 10.5	0.99% 12.5	43.00% 10.5	92.00% 2.5	56.98% 2.5	42.45% 10.5	92.98% 2.5	58.29% 2.5	2.5
Multi+$_{Mm}$	56.19% 10.5	0.99% 12.5	43.00% 10.5	92.00% 2.5	56.98% 2.5	42.45% 10.5	92.98% 2.5	58.29% 2.5	2.5
Multi+$_{MM}$	56.19% 10.5	0.99% 12.5	43.00% 10.5	92.00% 2.5	56.98% 2.5	42.45% 10.5	92.98% 2.5	58.29% 2.5	2.5
Multi±$_{stop,mm}$	75.58% 4.5	24.75% 4.5	58.42% 4.5	53.30% 12.5	52.57% 8.5	66.78% 4.5	51.38% 12.5	58.07% 8.5	8.5
Multi±$_{stop,mM}$	75.58% 4.5	24.75% 4.5	58.42% 4.5	53.30% 12.5	52.57% 8.5	66.78% 4.5	51.38% 12.5	58.07% 8.5	8.5
Multi±$_{stop,Mm}$	75.58% 4.5	24.75% 4.5	58.42% 4.5	53.30% 12.5	52.57% 8.5	66.78% 4.5	51.38% 12.5	58.07% 8.5	8.5
Multi±$_{stop,MM}$	75.58% 4.5	24.75% 4.5	58.42% 4.5	53.30% 12.5	52.57% 8.5	66.78% 4.5	51.38% 12.5	58.07% 8.5	8.5
Multi±$_{mm}$	75.58% 4.5	24.75% 4.5	58.42% 4.5	53.30% 12.5	52.57% 8.5	66.78% 4.5	51.38% 12.5	58.07% 8.5	8.5
Multi±$_{mM}$	75.58% 4.5	24.75% 4.5	58.42% 4.5	53.30% 12.5	52.57% 8.5	66.78% 4.5	51.38% 12.5	58.07% 8.5	8.5
Multi±$_{Mm}$	75.58% 4.5	24.75% 4.5	58.42% 4.5	53.30% 12.5	52.57% 8.5	66.78% 4.5	51.38% 12.5	58.07% 8.5	8.5
Multi±$_{MM}$	75.58% 4.5	24.75% 4.5	58.42% 4.5	53.30% 12.5	52.57% 8.5	66.78% 4.5	51.38% 12.5	58.07% 8.5	8.5
	Precision								
Multi+$_{stop,mm}$	68.89% 5.5	20.79% 1.5	49.65% 5.5	48.43% 11.5	46.80% 11.5	52.99% 5.5	48.87% 11.5	50.85% 7.5	7.5
Multi+$_{stop,mM}$	68.89% 5.5	20.79% 1.5	49.65% 5.5	48.43% 11.5	46.80% 11.5	52.99% 5.5	48.87% 11.5	50.85% 7.5	7.5
Multi+$_{stop,Mm}$	32.92% 14.5	0.00% 12.5	32.92% 10.5	100.00% 2.5	48.48% 8.5	32.92% 10.5	100.00% 2.5	49.53% 10.5	10.5
Multi+$_{stop,MM}$	32.92% 14.5	0.00% 12.5	32.92% 10.5	100.00% 2.5	48.48% 8.5	32.92% 10.5	100.00% 2.5	49.53% 10.5	10.5
Multi+$_{mm}$	47.28% 11.5	4.46% 7.5	38.77% 7.5	88.53% 5.5	50.89% 1.5	37.26% 7.5	87.97% 5.5	52.35% 1.5	5.5
Multi+$_{mM}$	47.28% 11.5	4.46% 7.5	38.77% 7.5	88.53% 5.5	50.89% 1.5	37.26% 7.5	87.97% 5.5	52.35% 1.5	5.5
Multi+$_{Mm}$	32.92% 14.5	0.00% 12.5	32.92% 10.5	100.00% 2.5	48.48% 8.5	32.92% 10.5	100.00% 2.5	49.53% 10.5	10.5
Multi+$_{MM}$	32.92% 14.5	0.00% 12.5	32.92% 10.5	100.00% 2.5	48.48% 8.5	32.92% 10.5	100.00% 2.5	49.53% 10.5	10.5
Multi±$_{stop,mm}$	69.88% 2.5	16.83% 4.5	51.07% 2.5	51.90% 8.5	49.11% 4.5	54.67% 2.5	49.87% 8.5	52.16% 4.5	2.5
Multi±$_{stop,mM}$	69.88% 2.5	16.83% 4.5	51.07% 2.5	51.90% 8.5	49.11% 4.5	54.67% 2.5	49.87% 8.5	52.16% 4.5	2.5
Multi±$_{stop,Mm}$	67.08% 8.5	0.00% 12.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	14.5
Multi±$_{stop,MM}$	67.08% 8.5	0.00% 12.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	14.5
Multi±$_{mm}$	69.88% 2.5	16.83% 4.5	51.07% 2.5	51.90% 8.5	49.11% 4.5	54.67% 2.5	49.87% 8.5	52.16% 4.5	2.5
Multi±$_{mM}$	69.88% 2.5	16.83% 4.5	51.07% 2.5	51.90% 8.5	49.11% 4.5	54.67% 2.5	49.87% 8.5	52.16% 4.5	2.5
Multi±$_{Mm}$	67.08% 8.5	0.00% 12.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	14.5
Multi±$_{MM}$	67.08% 8.5	0.00% 12.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	0.00% 14.5	14.5
	Subset Accuracy								
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	

Table 18: Predictive performance of different multi-label classification approaches on the data set EMOTIONS using the rule-independent evaluation strategy.

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
F-Measure									
Multi+$_{stop,mm}$	77.86% 3	31.66% 3.5	41.25% 3	97.57% 4	45.95% 3	16.12% 3	92.65% 7	27.47% 3	3.5
Multi+$_{stop,mM}$	59.30% 4	31.66% 3.5	36.58% 4	98.53% 3	40.26% 4	9.71% 4	95.51% 3	17.63% 4	3.5
Multi+$_{stop,Mm}$	22.93% 7.5	0.00% 6.5	5.50% 7.5	99.16% 1.5	10.28% 7.5	5.50% 7.5	98.37% 1.5	10.43% 7.5	7.5
Multi+$_{stop,MM}$	22.93% 7.5	0.00% 6.5	5.50% 7.5	99.16% 1.5	10.28% 7.5	5.50% 7.5	98.37% 1.5	10.43% 7.5	7.5
Multi+$_{mm}$	98.38% 1.5	74.87% 1.5	89.36% 1.5	97.11% 7.5	90.32% 1.5	76.69% 1.5	92.65% 7	83.92% 1.5	1.5
Multi+$_{mM}$	98.38% 1.5	74.87% 1.5	89.36% 1.5	97.11% 7.5	90.32% 1.5	76.69% 1.5	92.65% 7	83.92% 1.5	1.5
Multi+$_{Mm}$	52.08% 5.5	0.00% 6.5	8.18% 5.5	97.19% 5.5	14.86% 5.5	8.18% 5.5	93.06% 4.5	15.04% 5.5	5.5
Multi+$_{MM}$	52.08% 5.5	0.00% 6.5	8.18% 5.5	97.19% 5.5	14.86% 5.5	8.18% 5.5	93.06% 4.5	15.04% 5.5	5.5
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Hamming Accuracy									
Multi+$_{stop,mm}$	85.32% 6.5	39.70% 6.5	50.76% 6.5	99.16% 2.5	57.68% 6.5	23.49% 6.5	98.37% 2.5	37.92% 6.5	6.5
Multi+$_{stop,mM}$	85.32% 6.5	39.70% 6.5	50.76% 6.5	99.16% 2.5	57.68% 6.5	23.49% 6.5	98.37% 2.5	37.92% 6.5	6.5
Multi+$_{stop,Mm}$	85.32% 6.5	39.70% 6.5	50.76% 6.5	99.16% 2.5	57.68% 6.5	23.49% 6.5	98.37% 2.5	37.92% 6.5	6.5
Multi+$_{stop,MM}$	85.32% 6.5	39.70% 6.5	50.76% 6.5	99.16% 2.5	57.68% 6.5	23.49% 6.5	98.37% 2.5	37.92% 6.5	6.5
Multi+$_{mm}$	98.90% 2.5	77.39% 2.5	87.77% 2.5	92.84% 6.5	88.54% 2.5	87.20% 2.5	88.98% 6.5	88.08% 2.5	2.5
Multi+$_{mM}$	98.90% 2.5	77.39% 2.5	87.77% 2.5	92.84% 6.5	88.54% 2.5	87.20% 2.5	88.98% 6.5	88.08% 2.5	2.5
Multi+$_{Mm}$	98.90% 2.5	77.39% 2.5	87.77% 2.5	92.84% 6.5	88.54% 2.5	87.20% 2.5	88.98% 6.5	88.08% 2.5	2.5
Multi+$_{MM}$	98.90% 2.5	77.39% 2.5	87.77% 2.5	92.84% 6.5	88.54% 2.5	87.20% 2.5	88.98% 6.5	88.08% 2.5	2.5
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Precision									
Multi+$_{stop,mm}$	98.31% 1.5	73.87% 1.5	75.38% 1.5	74.54% 7.5	74.79% 1.5	98.73% 1.5	63.67% 7.5	77.42% 1.5	1.5
Multi+$_{stop,mM}$	98.31% 1.5	73.87% 1.5	75.38% 1.5	74.54% 7.5	74.79% 1.5	98.73% 1.5	63.67% 7.5	77.42% 1.5	1.5
Multi+$_{stop,Mm}$	22.93% 7.5	0.00% 6.5	5.50% 7.5	99.16% 1.5	10.28% 7.5	5.50% 7.5	98.37% 1.5	10.43% 7.5	7.5
Multi+$_{stop,MM}$	22.93% 7.5	0.00% 6.5	5.50% 7.5	99.16% 1.5	10.28% 7.5	5.50% 7.5	98.37% 1.5	10.43% 7.5	7.5
Multi+$_{mm}$	98.25% 3.5	71.86% 3.5	74.62% 3.5	75.04% 5.5	74.44% 3.5	95.76% 3.5	64.49% 5.5	77.07% 3.5	3.5
Multi+$_{mM}$	98.25% 3.5	71.86% 3.5	74.62% 3.5	75.04% 5.5	74.44% 3.5	95.76% 3.5	64.49% 5.5	77.07% 3.5	3.5
Multi+$_{Mm}$	52.08% 5.5	0.00% 6.5	8.18% 5.5	97.19% 3.5	14.86% 5.5	8.18% 5.5	93.06% 3.5	15.04% 5.5	5.5
Multi+$_{MM}$	52.08% 5.5	0.00% 6.5	8.18% 5.5	97.19% 3.5	14.86% 5.5	8.18% 5.5	93.06% 3.5	15.04% 5.5	5.5
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Subset Accuracy									
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	

Table 19: Predictive performance of different multi-label classification approaches on the data set GENBASE using the rule-independent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
	F-Measure								
Multi+$_{stop,mm}$	22.83% 5	0.00% 4.5	5.99% 6	52.09% 3	10.38% 6	6.08% 3	97.76% 3	11.44% 3	3.5
Multi+$_{stop,mM}$	13.65% 6	0.00% 4.5	7.03% 3	49.05% 5	10.96% 5	4.97% 8	87.86% 6	9.40% 8	8
Multi+$_{stop,Mm}$	5.10% 7.5	0.00% 4.5	5.10% 7.5	53.25% 1.5	9.13% 7.5	5.10% 6.5	100.00% 1.5	9.71% 6.5	6.5
Multi+$_{stop,MM}$	5.10% 7.5	0.00% 4.5	5.10% 7.5	53.25% 1.5	9.13% 7.5	5.10% 6.5	100.00% 1.5	9.71% 6.5	6.5
Multi+$_{mm}$	31.22% 3	0.00% 4.5	7.25% 2	45.85% 7	11.64% 3	5.83% 5	82.43% 8	10.89% 5	5
Multi+$_{mM}$	31.12% 4	0.00% 4.5	7.42% 1	45.60% 8	11.89% 1	5.98% 4	84.98% 7	11.18% 4	3.5
Multi+$_{Mm}$	31.61% 2	0.00% 4.5	6.48% 5	49.07% 4	11.16% 4	6.49% 2	93.29% 4	12.13% 2	2
Multi+$_{MM}$	36.14% 1	0.00% 4.5	6.83% 4	48.27% 6	11.66% 2	6.85% 1	91.37% 5	12.74% 1	1
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
	Hamming Accuracy								
Multi+$_{stop,mm}$	64.62% 2.5	10.22% 2.5	16.72% 2.5	47.50% 2.5	19.73% 2.5	10.33% 2.5	77.32% 4.5	18.23% 2.5	2.5
Multi+$_{stop,mM}$	64.62% 2.5	10.22% 2.5	16.72% 2.5	47.50% 2.5	19.73% 2.5	10.33% 2.5	77.32% 4.5	18.23% 2.5	2.5
Multi+$_{stop,Mm}$	64.62% 2.5	10.22% 2.5	16.72% 2.5	47.50% 2.5	19.73% 2.5	10.33% 2.5	77.32% 4.5	18.23% 2.5	2.5
Multi+$_{stop,MM}$	64.62% 2.5	10.22% 2.5	16.72% 2.5	47.50% 2.5	19.73% 2.5	10.33% 2.5	77.32% 4.5	18.23% 2.5	2.5
Multi+$_{mm}$	63.92% 6.5	1.24% 6.5	6.56% 6.5	38.44% 6.5	10.07% 6.5	10.15% 6.5	77.32% 4.5	17.94% 6.5	6.5
Multi+$_{mM}$	63.92% 6.5	1.24% 6.5	6.56% 6.5	38.44% 6.5	10.07% 6.5	10.15% 6.5	77.32% 4.5	17.94% 6.5	6.5
Multi+$_{Mm}$	63.92% 6.5	1.24% 6.5	6.56% 6.5	38.44% 6.5	10.07% 6.5	10.15% 6.5	77.32% 4.5	17.94% 6.5	6.5
Multi+$_{MM}$	63.92% 6.5	1.24% 6.5	6.56% 6.5	38.44% 6.5	10.07% 6.5	10.15% 6.5	77.32% 4.5	17.94% 6.5	6.5
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
	Precision								
Multi+$_{stop,mm}$	90.34% 1.5	4.33% 1.5	15.27% 1.5	14.78% 7.5	13.68% 1.5	17.44% 1.5	23.96% 7.5	20.19% 1.5	1.5
Multi+$_{stop,mM}$	90.34% 1.5	4.33% 1.5	15.27% 1.5	14.78% 7.5	13.68% 1.5	17.44% 1.5	23.96% 7.5	20.19% 1.5	1.5
Multi+$_{stop,Mm}$	5.10% 7.5	0.00% 6.5	5.10% 7.5	53.25% 1.5	9.13% 7.5	5.10% 7.5	100.00% 1.5	9.71% 7.5	7.5
Multi+$_{stop,MM}$	5.10% 7.5	0.00% 6.5	5.10% 7.5	53.25% 1.5	9.13% 7.5	5.10% 7.5	100.00% 1.5	9.71% 7.5	7.5
Multi+$_{mm}$	70.46% 3.5	2.48% 3.5	10.93% 3.5	26.60% 5.5	13.41% 3.5	7.96% 3.5	45.37% 5.5	13.54% 3.5	3.5
Multi+$_{mM}$	70.46% 3.5	2.48% 3.5	10.93% 3.5	26.60% 5.5	13.41% 3.5	7.96% 3.5	45.37% 5.5	13.54% 3.5	3.5
Multi+$_{Mm}$	36.14% 5.5	0.00% 6.5	6.83% 5.5	48.27% 3.5	11.66% 5.5	6.85% 5.5	91.37% 3.5	12.74% 5.5	5.5
Multi+$_{MM}$	36.14% 5.5	0.00% 6.5	6.83% 5.5	48.27% 3.5	11.66% 5.5	6.85% 5.5	91.37% 3.5	12.74% 5.5	5.5
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
	Subset Accuracy								
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	

Table 20: Predictive performance of different multi-label classification approaches on the data set BIRDS using the rule-independent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	Hamm. Acc.	Subset Acc.	Ex.-based Prec.	Ex.-based Rec.	Ex.-based F1	Mi. Prec.	Mi. Rec	Mi. F1	Avg. Rank
F-Measure									
Multi+$_{stop,mm}$	19.59% 11	0.00% 9.5	18.43% 3	99.83% 5.5	30.84% 3	18.35% 3	99.77% 6	31.00% 3	3
Multi+$_{stop,mM}$	19.37% 12	0.00% 9.5	18.40% 4	99.83% 5.5	30.79% 4	18.32% 4	99.85% 5	30.95% 4	4
Multi+$_{stop,Mm}$	18.10% 14.5	0.00% 9.5	18.10% 6.5	100.00% 2.5	30.42% 6.5	18.10% 6.5	100.00% 2.5	30.65% 6.5	6.5
Multi+$_{stop,MM}$	18.10% 14.5	0.00% 9.5	18.10% 6.5	100.00% 2.5	30.42% 6.5	18.10% 6.5	100.00% 2.5	30.65% 6.5	6.5
Multi+$_{mm}$	62.70% 9	6.19% 2	34.60% 2	84.78% 8	46.30% 2	30.59% 2	83.60% 8	44.79% 2	2
Multi+$_{mM}$	62.28% 10	6.61% 1	34.64% 1	87.00% 7	47.13% 1	30.72% 1	86.37% 7	45.32% 1	1
Multi+$_{Mm}$	18.10% 14.5	0.00% 9.5	18.10% 6.5	100.00% 2.5	30.42% 6.5	18.10% 6.5	100.00% 2.5	30.65% 6.5	6.5
Multi+$_{MM}$	18.10% 14.5	0.00% 9.5	18.10% 6.5	100.00% 2.5	30.42% 6.5	18.10% 6.5	100.00% 2.5	30.65% 6.5	6.5
Multi±$_{stop,mm}$	81.90% 4.5	0.00% 9.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	12.5
Multi±$_{stop,mM}$	81.90% 4.5	0.00% 9.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	12.5
Multi±$_{stop,Mm}$	81.90% 4.5	0.00% 9.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	12.5
Multi±$_{stop,MM}$	81.90% 4.5	0.00% 9.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	12.5
Multi±$_{mm}$	81.90% 4.5	0.00% 9.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	12.5
Multi±$_{mM}$	81.90% 4.5	0.00% 9.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	12.5
Multi±$_{Mm}$	81.90% 4.5	0.00% 9.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	12.5
Multi±$_{MM}$	81.90% 4.5	0.00% 9.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	0.00% 12.5	12.5
Hamming Accuracy									
Multi+$_{stop,mm}$	62.64% 14.5	9.28% 10.5	28.08% 14.5	60.74% 6.5	35.47% 14.5	26.69% 14.5	60.89% 6.5	37.11% 14.5	14.5
Multi+$_{stop,mM}$	62.64% 14.5	9.28% 10.5	28.08% 14.5	60.74% 6.5	35.47% 14.5	26.69% 14.5	60.89% 6.5	37.11% 14.5	14.5
Multi+$_{stop,Mm}$	62.64% 14.5	9.28% 10.5	28.08% 14.5	60.74% 6.5	35.47% 14.5	26.69% 14.5	60.89% 6.5	37.11% 14.5	14.5
Multi+$_{stop,MM}$	62.64% 14.5	9.28% 10.5	28.08% 14.5	60.74% 6.5	35.47% 14.5	26.69% 14.5	60.89% 6.5	37.11% 14.5	14.5
Multi+$_{mm}$	65.43% 10.5	7.69% 14.5	35.66% 10.5	82.19% 2.5	47.14% 10.5	32.15% 10.5	81.91% 2.5	46.17% 10.5	10.5
Multi+$_{mM}$	65.43% 10.5	7.69% 14.5	35.66% 10.5	82.19% 2.5	47.14% 10.5	32.15% 10.5	81.91% 2.5	46.17% 10.5	10.5
Multi+$_{Mm}$	65.43% 10.5	7.69% 14.5	35.66% 10.5	82.19% 2.5	47.14% 10.5	32.15% 10.5	81.91% 2.5	46.17% 10.5	10.5
Multi+$_{MM}$	65.43% 10.5	7.69% 14.5	35.66% 10.5	82.19% 2.5	47.14% 10.5	32.15% 10.5	81.91% 2.5	46.17% 10.5	10.5
Multi±$_{stop,mm}$	84.03% 4.5	46.07% 4.5	51.84% 4.5	48.95% 12.5	49.92% 4.5	57.04% 4.5	47.73% 12.5	51.97% 4.5	4.5
Multi±$_{stop,mM}$	84.03% 4.5	46.07% 4.5	51.84% 4.5	48.95% 12.5	49.92% 4.5	57.04% 4.5	47.73% 12.5	51.97% 4.5	4.5
Multi±$_{stop,Mm}$	84.03% 4.5	46.07% 4.5	51.84% 4.5	48.95% 12.5	49.92% 4.5	57.04% 4.5	47.73% 12.5	51.97% 4.5	4.5
Multi±$_{stop,MM}$	84.03% 4.5	46.07% 4.5	51.84% 4.5	48.95% 12.5	49.92% 4.5	57.04% 4.5	47.73% 12.5	51.97% 4.5	4.5
Multi±$_{mm}$	84.03% 4.5	46.07% 4.5	51.84% 4.5	48.95% 12.5	49.92% 4.5	57.04% 4.5	47.73% 12.5	51.97% 4.5	4.5
Multi±$_{mM}$	84.03% 4.5	46.07% 4.5	51.84% 4.5	48.95% 12.5	49.92% 4.5	57.04% 4.5	47.73% 12.5	51.97% 4.5	4.5
Multi±$_{Mm}$	84.03% 4.5	46.07% 4.5	51.84% 4.5	48.95% 12.5	49.92% 4.5	57.04% 4.5	47.73% 12.5	51.97% 4.5	4.5
Multi±$_{MM}$	84.03% 4.5	46.07% 4.5	51.84% 4.5	48.95% 12.5	49.92% 4.5	57.04% 4.5	47.73% 12.5	51.97% 4.5	4.5
Precision									
Multi+$_{stop,mm}$	79.65% 5.5	36.54% 1.5	40.89% 1.5	40.05% 7.5	40.04% 1.5	43.09% 1.5	38.65% 7.5	40.75% 1.5	1.5
Multi+$_{stop,mM}$	79.65% 5.5	36.54% 1.5	40.89% 1.5	40.05% 7.5	40.04% 1.5	43.09% 1.5	38.65% 7.5	40.75% 1.5	1.5
Multi+$_{stop,Mm}$	18.10% 10.5	0.00% 8.5	18.10% 6.5	100.00% 2.5	30.42% 6.5	18.10% 6.5	100.00% 2.5	30.65% 6.5	6.5
Multi+$_{stop,MM}$	18.10% 10.5	0.00% 8.5	18.10% 6.5	100.00% 2.5	30.42% 6.5	18.10% 6.5	100.00% 2.5	30.65% 6.5	6.5
Multi+$_{mm}$	47.91% 7.5	4.60% 3.5	24.54% 3.5	77.80% 5.5	34.58% 3.5	22.37% 3.5	75.98% 5.5	34.56% 3.5	3.5
Multi+$_{mM}$	47.91% 7.5	4.60% 3.5	24.54% 3.5	77.80% 5.5	34.58% 3.5	22.37% 3.5	75.98% 5.5	34.56% 3.5	3.5
Multi+$_{Mm}$	18.10% 10.5	0.00% 8.5	18.10% 6.5	100.00% 2.5	30.42% 6.5	18.10% 6.5	100.00% 2.5	30.65% 6.5	6.5
Multi+$_{MM}$	18.10% 10.5	0.00% 8.5	18.10% 6.5	100.00% 2.5	30.42% 6.5	18.10% 6.5	100.00% 2.5	30.65% 6.5	6.5
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{stop,Mm}$	81.90% 2.5	0.00% 8.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	10.5
Multi±$_{stop,MM}$	81.90% 2.5	0.00% 8.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	10.5
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Multi±$_{Mm}$	81.90% 2.5	0.00% 8.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	10.5
Multi±$_{MM}$	81.90% 2.5	0.00% 8.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	0.00% 10.5	10.5
Subset Accuracy									
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	

Table 21: Predictive performance of different multi-label classification approaches on the data set SCENE using the rule-independent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

B Results of Model Analyses

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
	F-Measure								
Single+$_{stop}$	102	75	2	0.98%	0.98%	98.04%	0	0.00%	1.00
Single+	56	0	1	1.79%	0.00%	98.21%	0	0.00%	1.00
Multi+$_{stop,mm}$	101	70	2	0.99%	0.99%	98.02%	0	0.00%	1.00
Multi+$_{stop,mM}$	101	70	2	0.99%	0.99%	98.02%	0	0.00%	1.00
Multi+$_{stop,Mm}$	18	1	0	0.00%	0.00%	100.00%	6	33.00%	2.11
Multi+$_{stop,MM}$	18	1	0	0.00%	0.00%	100.00%	6	33.00%	2.11
Multi+$_{mm}$	65	0	2	3.08%	0.00%	96.92%	0	0.00%	1.00
Multi+$_{mM}$	65	0	2	3.08%	0.00%	96.92%	0	0.00%	1.00
Multi+$_{Mm}$	15	0	0	0.00%	0.00%	100.00%	3	20.00%	1.40
Multi+$_{MM}$	15	0	0	0.00%	0.00%	100.00%	3	20.00%	1.40
Single±$_{stop}$	116	7	12	2.59%	5.17%	92.24%	0	0.00%	1.00
Single±	110	0	11	1.82%	5.45%	92.73%	0	0.00%	1.00
Multi±$_{stop,mm}$	164	8	19	4.88%	5.49%	89.63%	3	1.83%	1.05
Multi±$_{stop,mM}$	164	8	19	4.88%	5.49%	89.63%	3	1.83%	1.05
Multi±$_{stop,Mm}$	19	1	0	0.00%	0.00%	100.00%	7	36.84%	2.37
Multi±$_{stop,MM}$	19	1	0	0.00%	0.00%	100.00%	7	36.84%	2.37
Multi±$_{mm}$	157	0	16	3.82%	5.10%	91.08%	3	1.91%	1.06
Multi±$_{mM}$	157	0	16	3.82%	5.10%	91.08%	3	1.91%	1.06
Multi±$_{Mm}$	19	0	0	0.00%	0.00%	100.00%	7	36.84%	2.37
Multi±$_{MM}$	19	0	0	0.00%	0.00%	100.00%	7	36.84%	2.37
	Hamming Accuracy								
Single+$_{stop}$	83	50	1	1.20%	0.00%	98.80%	0	0.00%	1.00
Single+	64	0	3	3.13%	1.56%	95.31%	0	0.00%	1.00
Multi+$_{stop,mm}$	78	39	0	0.00%	0.00%	100.00%	1	1.28%	1.01
Multi+$_{stop,mM}$	78	39	0	0.00%	0.00%	100.00%	1	1.28%	1.01
Multi+$_{stop,Mm}$	78	39	0	0.00%	0.00%	100.00%	1	1.28%	1.01
Multi+$_{stop,MM}$	78	39	0	0.00%	0.00%	100.00%	1	1.28%	1.01
Multi+$_{mm}$	68	0	0	0.00%	0.00%	100.00%	1	1.47%	1.01
Multi+$_{mM}$	68	0	0	0.00%	0.00%	100.00%	1	1.47%	1.01
Multi+$_{Mm}$	68	0	0	0.00%	0.00%	100.00%	1	1.47%	1.01
Multi+$_{MM}$	68	0	0	0.00%	0.00%	100.00%	1	1.47%	1.01
Single±$_{stop}$	89	1	5	0.00%	4.49%	95.51%	0	0.00%	1.00
Single±	89	0	5	0.00%	4.49%	95.51%	0	0.00%	1.00
Multi±$_{stop,mm}$	111	12	4	0.00%	2.70%	97.30%	2	1.80%	1.09
Multi±$_{stop,mM}$	111	12	4	0.00%	2.70%	97.30%	2	1.80%	1.09
Multi±$_{stop,Mm}$	111	12	4	0.00%	2.70%	97.30%	2	1.80%	1.09
Multi±$_{stop,MM}$	111	12	4	0.00%	2.70%	97.30%	2	1.80%	1.09
Multi±$_{mm}$	111	0	4	0.00%	2.70%	97.30%	2	1.80%	1.09
Multi±$_{mM}$	111	0	4	0.00%	2.70%	97.30%	2	1.80%	1.09
Multi±$_{Mm}$	111	0	4	0.00%	2.70%	97.30%	2	1.80%	1.09
Multi±$_{MM}$	111	0	4	0.00%	2.70%	97.30%	2	1.80%	1.09
	Precision								
Single+$_{stop}$	266	196	7	1.50%	1.13%	97.37%	0	0.00%	1.00
Single+	192	0	6	3.10%	0.00%	96.90%	0	0.00%	1.00
Multi+$_{stop,mm}$	227	203	0	0.00%	0.00%	100.00%	48	21.15%	1.22
Multi+$_{stop,mM}$	227	203	0	0.00%	0.00%	100.00%	48	21.15%	1.22
Multi+$_{stop,Mm}$	18	1	0	0.00%	0.00%	100.00%	6	33.33%	2.11
Multi+$_{stop,MM}$	18	1	0	0.00%	0.00%	100.00%	6	33.33%	2.11
Multi+$_{mm}$	177	0	4	2.26%	0.00%	97.74%	44	24.86%	1.27
Multi+$_{mM}$	177	0	4	2.26%	0.00%	97.74%	44	24.86%	1.27
Multi+$_{Mm}$	15	0	0	0.00%	0.00%	100.00%	3	20.00%	1.40
Multi+$_{MM}$	15	0	0	0.00%	0.00%	100.00%	3	20.00%	1.40
Single±$_{stop}$	408	10	75	18.38%	0.00%	81.62%	0	0.00%	1.00
Single±	403	0	75	18.61%	0.00%	81.39%	0	0.00%	1.00
Multi±$_{stop,mm}$	57	14	8	5.26%	8.77%	85.97%	53	92.98%	13.67
Multi±$_{stop,mM}$	57	14	8	5.26%	8.77%	85.97%	53	92.98%	13.67
Multi±$_{stop,Mm}$	19	1	0	0.00%	0.00%	100.00%	7	36.84%	2.37
Multi±$_{stop,MM}$	19	1	0	0.00%	0.00%	100.00%	7	36.84%	2.37
Multi±$_{mm}$	53	0	7	3.77%	9.43%	86.80%	51	96.23%	14.55
Multi±$_{mM}$	53	0	7	3.77%	9.43%	86.80%	51	96.23%	14.55
Multi±$_{Mm}$	19	0	0	0.00%	0.00%	100.00%	7	36.84%	2.37
Multi±$_{MM}$	19	0	0	0.00%	0.00%	100.00%	7	36.84%	2.37
	Subset Accuracy								
Multi+$_{stop,Mm}$	82	44	1	1.22%	0.00%	98.78%	3	3.66%	1.04
Multi+$_{Mm}$	68	0	2	2.94%	0.00%	97.06%	3	4.41%	1.04
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Table 22: Model analysis of different multi-label classification approaches on the data set MEDICAL using the rule-dependent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
	F-Measure								
Single+$_{stop}$	106	50	1	0.00%	0.94%	99.06%	0	0.00%	1.00
Single+	36	0	6	2.86%	14.29%	82.85%	0	0.00%	1.00
Multi+$_{stop,mm}$	115	54	2	0.00%	1.74%	98.26%	0	0.00%	1.00
Multi+$_{stop,mM}$	115	54	2	0.00%	1.74%	98.26%	0	0.00%	1.00
Multi+$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	41	0	2	0.00%	4.88%	95.12%	0	0.00%	1.00
Multi+$_{mM}$	41	0	2	0.00%	4.88%	95.12%	0	0.00%	1.00
Multi+$_{Mm}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single±$_{stop}$	48	5	9	0.00%	16.67%	83.33%	0	0.00%	1.00
Single±	48	0	9	0.00%	16.67%	83.33%	0	0.00%	1.00
Multi±$_{stop,mm}$	61	6	8	0.00%	11.48%	88.52%	0	0.00%	1.00
Multi±$_{stop,mM}$	61	6	8	0.00%	11.48%	88.52%	0	0.00%	1.00
Multi±$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{mm}$	61	0	9	0.00%	13.11%	86.89%	0	0.00%	1.00
Multi±$_{mM}$	61	0	9	0.00%	13.11%	86.89%	0	0.00%	1.00
Multi±$_{Mm}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{MM}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
	Hamming Accuracy								
Single+$_{stop}$	97	41	1	0.00%	1.03%	98.97%	0	0.00%	1.00
Single+	69	0	3	1.45%	2.90%	95.65%	0	0.00%	1.00
Multi+$_{stop,mm}$	205	93	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,mM}$	205	93	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,Mm}$	205	93	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	205	93	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	107	0	3	0.00%	2.80%	97.20%	0	0.00%	1.00
Multi+$_{mM}$	107	0	3	0.00%	2.80%	97.20%	0	0.00%	1.00
Multi+$_{Mm}$	107	0	3	0.00%	2.80%	97.20%	0	0.00%	1.00
Multi+$_{MM}$	107	0	3	0.00%	2.80%	97.20%	0	0.00%	1.00
Single±$_{stop}$	20	2	2	0.00%	10.00%	90.00%	0	0.00%	1.00
Single±	20	0	2	0.00%	10.00%	90.00%	0	0.00%	1.00
Multi±$_{stop,mm}$	51	11	3	0.00%	3.92%	96.08%	0	0.00%	1.00
Multi±$_{stop,mM}$	51	0	3	0.00%	3.92%	96.08%	0	0.00%	1.00
Multi±$_{stop,Mm}$	51	0	3	0.00%	3.92%	96.08%	0	0.00%	1.00
Multi±$_{stop,MM}$	51	0	3	0.00%	3.92%	96.08%	0	0.00%	1.00
Multi±$_{mm}$	51	0	3	0.00%	3.92%	96.08%	0	0.00%	1.00
Multi±$_{mM}$	51	0	3	0.00%	3.92%	96.08%	0	0.00%	1.00
Multi±$_{Mm}$	51	0	3	0.00%	3.92%	96.08%	0	0.00%	1.00
Multi±$_{MM}$	51	0	3	0.00%	3.92%	96.08%	0	0.00%	1.00
	Precision								
Single+$_{stop}$	355	171	1	0.00%	0.28%	99.72%	0	0.00%	1.00
Single+	153	0	1	0.00%	0.65%	99.35%	0	0.00%	1.00
Multi+$_{stop,mm}$	268	201	0	0.00%	0.00%	100.00%	116	42.28%	1.50
Multi+$_{stop,mM}$	268	201	0	0.00%	0.00%	100.00%	116	42.28%	1.50
Multi+$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	135	0	0	0.00%	0.00%	100.00%	52	38.52%	1.44
Multi+$_{mM}$	135	0	0	0.00%	0.00%	100.00%	52	38.52%	1.44
Multi+$_{Mm}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single±$_{stop}$	262	9	11	4.20%	0.00%	95.80%	0	0.00%	1.00
Single±	245	0	16	6.53%	0.00%	93.47%	0	0.00%	1.00
Multi±$_{stop,mm}$	100	33	0	0.00%	0.00%	100.00%	98	98.00%	4.34
Multi±$_{stop,mM}$	100	33	0	0.00%	0.00%	100.00%	98	98.00%	4.34
Multi±$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{mm}$	94	0	0	0.00%	0.00%	100.00%	92	97.87%	4.59
Multi±$_{mM}$	94	0	0	0.00%	0.00%	100.00%	92	97.87%	4.59
Multi±$_{Mm}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{MM}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
	Subset Accuracy								
Multi+$_{stop,Mm}$	197	108	1	0.00%	0.51%	99.49%	0	0.00%	1.00
Multi+$_{Mm}$	109	0	3	0.00%	2.75%	97.25%	0	0.00%	1.00
Multi±$_{stop,Mm}$	108	16	1	0.00%	0.93%	99.07%	0	0.00%	1.00
Multi±$_{Mm}$	84	0	1	0.00%	1.19%	98.81%	0	0.00%	1.00

Table 23: Model analysis of different multi-label classification approaches on the data set EMOTIONS using the rule-dependent evaluation strategy.

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
	F-Measure								
Single+$_{stop}$	30	16	1	3.33%	0.00%	96.67%	0	0.00%	1.00
Single+	24	0	1	4.17%	0.00%	95.83%	0	0.00%	1.00
Multi+$_{stop,mm}$	32	17	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,mM}$	32	17	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,Mm}$	20	1	0	0.00%	0.00%	100.00%	1	5.00%	1.10
Multi+$_{stop,MM}$	20	1	0	0.00%	0.00%	100.00%	1	5.00%	1.10
Multi+$_{mm}$	24	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mM}$	24	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{Mm}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single±$_{stop}$	54	1	2	1.85%	1.85%	96.30%	0	0.00%	1.00
Single±	54	0	2	1.85%	1.85%	96.30%	0	0.00%	1.00
Multi±$_{stop,mm}$	56	3	1	1.79%	0.00%	98.21%	1	1.79%	1.02
Multi±$_{stop,mM}$	56	3	1	1.79%	0.00%	98.21%	1	1.79%	1.02
Multi±$_{stop,Mm}$	22	1	0	0.00%	0.00%	100.00%	3	13.64%	1.23
Multi±$_{stop,MM}$	22	1	0	0.00%	0.00%	100.00%	3	13.64%	1.23
Multi±$_{mm}$	53	0	1	1.89%	0.00%	98.11%	1	1.89%	1.02
Multi±$_{mM}$	53	0	1	1.89%	0.00%	98.11%	1	1.89%	1.02
Multi±$_{Mm}$	22	0	0	0.00%	0.00%	100.00%	3	13.64%	1.23
Multi±$_{MM}$	22	0	0	0.00%	0.00%	100.00%	3	13.64%	1.23
	Hamming Accuracy								
Single+$_{stop}$	29	15	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single+	23	0	2	8.70%	0.00%	91.30%	0	0.00%	1.00
Multi+$_{stop,mm}$	28	13	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,mM}$	28	13	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,Mm}$	28	13	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	28	13	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	24	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mM}$	24	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{Mm}$	24	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	24	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single±$_{stop}$	52	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single±	52	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,mm}$	54	3	3	0.00%	3.70%	96.30%	1	1.85%	1.02
Multi±$_{stop,mM}$	54	3	3	0.00%	3.70%	96.30%	1	1.85%	1.02
Multi±$_{stop,Mm}$	54	3	3	0.00%	3.70%	96.30%	1	1.85%	1.02
Multi±$_{stop,MM}$	54	3	3	0.00%	3.70%	96.30%	1	1.85%	1.02
Multi±$_{mm}$	54	0	3	0.00%	3.70%	96.30%	1	1.85%	1.02
Multi±$_{mM}$	54	0	3	0.00%	3.70%	96.30%	1	1.85%	1.02
Multi±$_{Mm}$	54	0	3	0.00%	3.70%	96.30%	1	1.85%	1.02
Multi±$_{MM}$	54	0	3	0.00%	3.70%	96.30%	1	1.85%	1.02
	Precision								
Single+$_{stop}$	53	38	1	1.89%	0.00%	98.11%	0	0.00%	1.00
Single+	15	0	1	6.67%	0.00%	93.33%	0	0.00%	1.00
Multi+$_{stop,mm}$	37	25	0	0.00%	0.00%	100.00%	5	13.51%	1.16
Multi+$_{stop,mM}$	37	25	0	0.00%	0.00%	100.00%	5	13.51%	1.16
Multi+$_{stop,Mm}$	20	1	0	0.00%	0.00%	100.00%	1	5.00%	1.10
Multi+$_{stop,MM}$	20	1	0	0.00%	0.00%	100.00%	1	5.00%	1.10
Multi+$_{mm}$	14	0	0	0.00%	0.00%	100.00%	4	28.57%	1.36
Multi+$_{mM}$	14	0	0	0.00%	0.00%	100.00%	4	28.57%	1.36
Multi+$_{Mm}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single±$_{stop}$	72	10	9	12.68%	0.00%	87.32%	0	0.00%	1.00
Single±	51	0	8	15.69%	0.00%	84.31%	0	0.00%	1.00
Multi±$_{stop,mm}$	23	9	4	0.00%	17.39%	82.61%	16	69.57%	12.57
Multi±$_{stop,mM}$	23	9	4	0.00%	17.39%	82.61%	16	69.57%	12.57
Multi±$_{stop,Mm}$	22	1	0	0.00%	0.00%	100.00%	3	13.64%	1.23
Multi±$_{stop,MM}$	22	1	0	0.00%	0.00%	100.00%	3	13.64%	1.23
Multi±$_{mm}$	15	0	4	0.00%	26.67%	73.33%	14	93.33%	18.40
Multi±$_{mM}$	15	0	4	0.00%	26.67%	73.33%	14	93.33%	18.40
Multi±$_{Mm}$	22	0	0	0.00%	0.00%	100.00%	3	13.64%	1.23
Multi±$_{MM}$	22	0	0	0.00%	0.00%	100.00%	3	13.64%	1.23
	Subset Accuracy								
Multi+$_{stop,Mm}$	28	13	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{Mm}$	24	0	6	25.00%	0.00%	75.00%	6	25.00%	1.25
Multi±$_{stop,Mm}$	54	2	6	5.56%	5.56%	88.88%	54	100.00%	3.30
Multi±$_{Mm}$	54	0	6	5.56%	5.56%	88.88%	54	100.00%	3.30

Table 24: Model analysis of different multi-label classification approaches on the data set GENBASE using the rule-dependent evaluation strategy.

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
F-Measure									
Single+ $_{stop}$	155	85	3	0.65%	1.29%	98.06%	0	0.00%	1.00
Single+	135	0	8	2.96%	2.96%	94.08%	0	0.00%	1.00
Multi+ $_{stop,mm}$	149	73	1	0.67%	0.00%	99.33%	2	1.34%	1.01
Multi+ $_{stop,mM}$	149	73	1	0.67%	0.00%	99.33%	2	1.34%	1.01
Multi+ $_{stop,Mm}$	14	1	0	0.00%	0.00%	100.00%	5	0.36%	1.36
Multi+ $_{stop,MM}$	14	1	0	0.00%	0.00%	100.00%	5	0.36%	1.36
Multi+ $_{mm}$	132	0	5	3.79%	0.00%	96.21%	2	1.52%	1.02
Multi+ $_{mM}$	132	0	5	3.79%	0.00%	96.21%	2	1.52%	1.02
Multi+ $_{Mm}$	11	0	0	0.00%	0.00%	100.00%	3	27.27%	1.27
Multi+ $_{MM}$	11	0	0	0.00%	0.00%	100.00%	3	27.27%	1.27
Single± $_{stop}$	81	4	2	0.00%	2.47%	97.53%	0	0.00%	1.00
Single±	80	0	2	0.00%	2.50%	97.50%	0	0.00%	1.00
Multi± $_{stop,mm}$	136	6	1	0.00%	0.74%	99.26%	0	0.00%	1.00
Multi± $_{stop,mM}$	136	6	1	0.00%	0.74%	99.26%	0	0.00%	1.00
Multi± $_{stop,Mm}$	16	2	0	0.00%	0.00%	100.00%	6	37.50%	1.56
Multi± $_{stop,MM}$	16	2	0	0.00%	0.00%	100.00%	6	37.50%	1.56
Multi± $_{mm}$	131	0	1	0.00%	0.76%	99.24%	0	0.00%	1.00
Multi± $_{mM}$	131	0	1	0.00%	0.76%	99.24%	0	0.00%	1.00
Multi± $_{Mm}$	14	0	0	0.00%	0.00%	100.00%	5	35.71%	1.36
Multi± $_{MM}$	14	0	0	0.00%	0.00%	100.00%	5	35.71%	1.36
Hamming Accuracy									
Single+ $_{stop}$	167	63	5	1.20%	18.00%	80.80%	0	0.00%	1.00
Single+	155	0	7	2.58%	1.94%	95.48%	0	0.00%	1.00
Multi+ $_{stop,mm}$	160	54	3	0.00%	1.88%	98.12%	0	0.00%	1.00
Multi+ $_{stop,mM}$	160	54	3	0.00%	1.88%	98.12%	0	0.00%	1.00
Multi+ $_{stop,Mm}$	160	54	3	0.00%	1.88%	98.12%	0	0.00%	1.00
Multi+ $_{stop,MM}$	160	54	3	0.00%	1.88%	98.12%	0	0.00%	1.00
Multi+ $_{mm}$	156	0	4	0.64%	1.92%	97.44%	0	0.00%	1.00
Multi+ $_{mM}$	156	0	4	0.64%	1.92%	97.44%	0	0.00%	1.00
Multi+ $_{Mm}$	156	0	4	0.64%	1.92%	97.44%	0	0.00%	1.00
Multi+ $_{MM}$	156	0	4	0.64%	1.92%	97.44%	0	0.00%	1.00
Single± $_{stop}$	48	2	1	0.00%	2.08%	97.92%	0	0.00%	1.00
Single±	48	0	1	0.00%	2.08%	97.92%	0	0.00%	1.00
Multi± $_{stop,mm}$	130	19	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Multi± $_{stop,mM}$	130	19	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Multi± $_{stop,Mm}$	130	19	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Multi± $_{stop,MM}$	130	19	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Multi± $_{mm}$	130	0	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Multi± $_{mM}$	130	0	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Multi± $_{Mm}$	130	0	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Multi± $_{MM}$	130	0	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Precision									
Single+ $_{stop}$	240	129	6	2.08%	0.42%	97.50%	0	0.00%	1.00
Single+	194	0	20	10.31%	0.00%	89.69%	0	0.00%	1.00
Multi+ $_{stop,mm}$	165	133	0	0.00%	0.00%	100.00%	58	35.15%	1.53
Multi+ $_{stop,mM}$	165	133	0	0.00%	0.00%	100.00%	58	35.15%	1.53
Multi+ $_{stop,Mm}$	14	1	0	0.00%	0.00%	100.00%	5	35.71%	1.36
Multi+ $_{stop,MM}$	14	1	0	0.00%	0.00%	100.00%	5	35.71%	1.36
Multi+ $_{mm}$	151	0	0	0.00%	0.00%	100.00%	55	36.42%	1.53
Multi+ $_{mM}$	151	0	0	0.00%	0.00%	100.00%	55	36.42%	1.53
Multi+ $_{Mm}$	11	0	0	0.00%	0.00%	100.00%	3	27.27%	1.27
Multi+ $_{MM}$	11	0	0	0.00%	0.00%	100.00%	3	27.27%	1.27
Single± $_{stop}$	227	1	5	2.20%	0.00%	97.80%	0	0.00%	1.00
Single±	210	0	5	2.38%	0.00%	97.62%	0	0.00%	1.00
Multi± $_{stop,mm}$	55	6	0	0.00%	0.00%	100.00%	55	100.00%	11.45
Multi± $_{stop,mM}$	55	6	0	0.00%	0.00%	100.00%	55	100.00%	11.45
Multi± $_{stop,Mm}$	16	2	0	0.00%	0.00%	100.00%	6	37.50%	1.56
Multi± $_{stop,MM}$	16	2	0	0.00%	0.00%	100.00%	6	37.50%	1.56
Multi± $_{mm}$	53	0	0	0.00%	0.00%	100.00%	53	100.00%	11.60
Multi± $_{mM}$	53	0	0	0.00%	0.00%	100.00%	53	100.00%	11.60
Multi± $_{Mm}$	14	0	0	0.00%	0.00%	100.00%	5	35.71%	35.71
Multi± $_{MM}$	14	0	0	0.00%	0.00%	100.00%	5	35.71%	35.71
Subset Accuracy									
Multi+ $_{stop,Mm}$	158	57	4	0.00%	2.53%	97.47%	0	0.00%	1.00
Multi+ $_{Mm}$	156	0	5	1.92%	1.28%	96.80%	5	3.21%	1.03
Multi± $_{stop,Mm}$	130	19	1	0.77%	0.00%	99.23%	0	0.00%	1.00
Multi± $_{Mm}$	130	0	1	0.77%	0.00%	99.23%	0	0.00%	1.00

Table 25: Model analysis of different multi-label classification approaches on the data set BIRDS using the rule-dependent evaluation strategy.

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
	F-Measure								
Single+$_{stop}$	152	119	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single+	30	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,mm}$	149	115	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,mM}$	149	115	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	45	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mM}$	45	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{Mm}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single±$_{stop}$	56	6	6	1.79%	5.36%	92.85%	0	0.00%	1.00
Single±	55	0	5	1.81%	5.45%	92.74%	0	0.00%	1.00
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	55	7	10	0.00%	10.91%	89.09%	0	0.00%	1.00
Multi±$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{mm}$	51	0	8	1.96%	9.80%	88.24%	0	0.00%	1.00
Multi±$_{mM}$	51	0	8	1.96%	9.80%	88.24%	0	0.00%	1.00
Multi±$_{Mm}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{MM}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
	Hamming Accuracy								
Single+$_{stop}$	251	213	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single+	114	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,mm}$	321	273	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,mM}$	321	273	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,Mm}$	321	273	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	321	273	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	148	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mM}$	148	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{Mm}$	148	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	148	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single±$_{stop}$	23	2	2	0.00%	8.70%	91.30%	0	0.00%	1.00
Single±	23	0	2	0.00%	8.70%	91.30%	0	0.00%	1.00
Multi±$_{stop,mm}$	58	10	3	5.17%	0.00%	94.83%	0	0.00%	1.00
Multi±$_{stop,mM}$	58	10	3	5.17%	0.00%	94.83%	0	0.00%	1.00
Multi±$_{stop,Mm}$	58	10	3	5.17%	0.00%	94.83%	0	0.00%	1.00
Multi±$_{stop,MM}$	58	10	3	5.17%	0.00%	94.83%	0	0.00%	1.00
Multi±$_{mm}$	58	0	3	5.17%	0.00%	94.83%	0	0.00%	1.00
Multi±$_{mM}$	58	0	3	5.17%	0.00%	94.83%	0	0.00%	1.00
Multi±$_{Mm}$	58	0	3	5.17%	0.00%	94.83%	0	0.00%	1.00
Multi±$_{MM}$	58	0	3	5.17%	0.00%	94.83%	0	0.00%	1.00
	Precision								
Single+$_{stop}$	483	408	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Single+	288	0	4	1.39%	0.00%	98.61%	0	0.00%	1.00
Multi+$_{stop,mm}$	443	384	0	0.00%	0.00%	100.00%	20	4.51%	1.05
Multi+$_{stop,mM}$	443	384	0	0.00%	0.00%	100.00%	20	4.51%	1.05
Multi+$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	326	0	1	0.31%	0.00%	99.69%	10	3.07%	1.03
Multi+$_{mM}$	326	0	1	0.31%	0.00%	99.69%	10	3.07%	1.03
Multi+$_{Mm}$	6	0	1	16.67%	0.00%	83.33%	0	0.00%	1.00
Multi+$_{MM}$	6	0	1	16.67%	0.00%	83.33%	0	0.00%	1.00
Single±$_{stop}$	358	21	29	8.10%	0.00%	91.90%	0	0.00%	1.00
Single±	338	0	25	7.40%	0.00%	92.60%	0	0.00%	1.00
Multi±$_{stop,mm}$	144	66	0	0.00%	0.00%	100.00%	143	99.31%	3.95
Multi±$_{stop,mM}$	144	66	0	0.00%	0.00%	100.00%	143	99.31%	3.95
Multi±$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{mm}$	133	0	0	0.00%	0.00%	100.00%	132	99.25%	4.02
Multi±$_{mM}$	133	0	0	0.00%	0.00%	100.00%	132	99.25%	4.02
Multi±$_{Mm}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{MM}$	6	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
	Subset Accuracy								
Multi+$_{stop,Mm}$	325	274	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{Mm}$	153	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,Mm}$	60	10	3	0.00%	5.00%	95.00%	0	0.00%	1.00
Multi±$_{Mm}$	60	0	3	0.00%	5.00%	95.00%	0	0.00%	1.00

Table 26: Model analysis of different multi-label classification approaches on the data set SCENE using the rule-dependent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
F-Measure									
Multi+$_{stop,mm}$	117	1	111	82.05%	8.55%	9.40%	47	40.17%	1.74
Multi+$_{stop,mM}$	86	2	7	8.14%	0.00%	91.86%	41	47.67%	1.51
Multi+$_{stop,Mm}$	35	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	35	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	27	0	15	25.93%	29.63%	44.44%	0	0.00%	1.00
Multi+$_{mM}$	20	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{Mm}$	20	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	20	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Hamming Accuracy									
Multi+$_{stop,mm}$	183	95	56	2.73%	27.32%	69.95%	67	36.61%	1.55
Multi+$_{stop,mM}$	183	95	56	2.73%	27.32%	69.95%	67	36.61%	1.55
Multi+$_{stop,Mm}$	183	95	56	2.73%	27.32%	69.95%	67	36.61%	1.55
Multi+$_{stop,MM}$	183	95	56	2.73%	27.32%	69.95%	67	36.61%	1.55
Multi+$_{mm}$	44	0	11	9.09%	15.90%	75.01%	6	13.64%	1.14
Multi+$_{mM}$	44	0	11	9.09%	15.90%	75.01%	6	13.64%	1.14
Multi+$_{Mm}$	44	0	11	9.09%	15.90%	75.01%	6	13.64%	1.14
Multi+$_{MM}$	44	0	11	9.09%	15.90%	75.01%	6	13.64%	1.14
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Precision									
Multi+$_{stop,mm}$	223	223	0	0.00%	0.00%	100.00%	58	26.01%	1.26
Multi+$_{stop,mM}$	223	223	0	0.00%	0.00%	100.00%	58	26.01%	1.26
Multi+$_{stop,Mm}$	35	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	35	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	223	0	0	0.00%	0.00%	100.00%	58	26.01%	1.26
Multi+$_{mM}$	223	0	0	0.00%	0.00%	100.00%	58	26.01%	1.26
Multi+$_{Mm}$	20	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	20	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Subset Accuracy									
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	-

Table 27: Model analysis of different multi-label classification approaches on the data set MEDICAL using the rule-independent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
	F-Measure								
Multi+$_{stop,mm}$	12	1	4	16.67%	16.67%	66.66%	7	58.33%	1.75
Multi+$_{stop,mM}$	16	1	5	31.25%	0.00%	68.75%	9	56.25%	1.69
Multi+$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	6	1	3	50.00%	0.00%	50.00%	0	0.00%	1.00
Multi+$_{mm}$	8	0	0	0.00%	0.00%	100.00%	5	62.50%	1.63
Multi+$_{mM}$	9	0	0	0.00%	0.00%	100.00%	3	33.33%	1.33
Multi+$_{Mm}$	5	0	2	40.00%	0.00%	60.00%	2	40.00%	1.60
Multi+$_{MM}$	5	0	1	20.00%	0.00%	80.00%	2	40.00%	1.60
Multi±$_{stop,mm}$	2	2	0	0.00%	0.00%	100.00%	2	100.00%	6.00
Multi±$_{stop,mM}$	2	2	0	0.00%	0.00%	100.00%	2	100.00%	6.00
Multi±$_{stop,Mm}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{stop,MM}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{mm}$	2	0	0	0.00%	0.00%	100.00%	2	100.00%	6.00
Multi±$_{mM}$	2	0	0	0.00%	0.00%	100.00%	2	100.00%	6.00
Multi±$_{Mm}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{MM}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
	Hamming Accuracy								
Multi+$_{stop,mm}$	86	29	34	0.00%	38.37%	61.63%	64	74.42%	1.91
Multi+$_{stop,mM}$	86	29	34	0.00%	38.37%	61.63%	64	74.42%	1.91
Multi+$_{stop,Mm}$	86	29	34	0.00%	38.37%	61.63%	64	74.42%	1.91
Multi+$_{stop,MM}$	86	29	34	0.00%	38.37%	61.63%	64	74.42%	1.91
Multi+$_{mm}$	17	0	1	0.00%	5.88%	94.12%	13	76.47%	1.94
Multi+$_{mM}$	17	0	1	0.00%	5.88%	94.12%	13	76.47%	1.94
Multi+$_{Mm}$	17	0	1	0.00%	5.88%	94.12%	13	76.47%	1.94
Multi+$_{MM}$	17	0	1	0.00%	5.88%	94.12%	13	76.47%	1.94
Multi±$_{stop,mm}$	6	6	0	0.00%	0.00%	100.00%	6	100.00%	6.00
Multi±$_{stop,mM}$	6	0	0	0.00%	0.00%	100.00%	6	100.00%	6.00
Multi±$_{stop,Mm}$	6	0	0	0.00%	0.00%	100.00%	6	100.00%	6.00
Multi±$_{stop,MM}$	6	0	0	0.00%	0.00%	100.00%	6	100.00%	6.00
Multi±$_{mm}$	6	0	0	0.00%	0.00%	100.00%	6	100.00%	6.00
Multi±$_{mM}$	6	0	0	0.00%	0.00%	100.00%	6	100.00%	6.00
Multi±$_{Mm}$	6	0	0	0.00%	0.00%	100.00%	6	100.00%	6.00
Multi±$_{MM}$	6	0	0	0.00%	0.00%	100.00%	6	100.00%	6.00
	Precision								
Multi+$_{stop,mm}$	266	245	0	0.00%	0.00%	100.00%	175	65.79%	1.76
Multi+$_{stop,mM}$	266	245	0	0.00%	0.00%	100.00%	175	65.79%	1.76
Multi+$_{stop,Mm}$	6	1	3	50.00%	0.00%	50.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	6	1	3	50.00%	0.00%	50.00%	0	0.00%	1.00
Multi+$_{mm}$	246	0	0	0.00%	0.00%	100.00%	155	63.01%	1.68
Multi+$_{mM}$	246	0	0	0.00%	0.00%	100.00%	155	63.01%	1.68
Multi+$_{Mm}$	5	0	1	20.00%	0.00%	80.00%	2	40.00%	1.60
Multi+$_{MM}$	5	0	1	20.00%	0.00%	80.00%	2	40.00%	1.60
Multi±$_{stop,mm}$	223	223	0	0.00%	0.00%	100.00%	223	100.00%	6.00
Multi±$_{stop,mM}$	223	223	0	0.00%	0.00%	100.00%	223	100.00%	6.00
Multi±$_{stop,Mm}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{stop,MM}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{mm}$	223	0	0	0.00%	0.00%	100.00%	223	100.00%	6.00
Multi±$_{mM}$	223	0	0	0.00%	0.00%	100.00%	223	100.00%	6.00
Multi±$_{Mm}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{MM}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
	Subset Accuracy								
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	-

Table 28: Model analysis of different multi-label classification approaches on the data set EMOTIONS using the rule-independent evaluation strategy.

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
F-Measure									
Multi+$_{stop,mm}$	24	4	7	16.67%	12.50%	70.83%	11	45.83%	2.25
Multi+$_{stop,mM}$	21	2	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,Mm}$	22	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	22	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	12	0	1	8.33%	0.00%	91.67%	3	25.00%	1.42
Multi+$_{mM}$	12	0	1	8.33%	0.00%	91.67%	3	25.00%	1.42
Multi+$_{Mm}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Hamming Accuracy									
Multi+$_{stop,mm}$	186	32	148	32.80%	34.41%	32.79%	138	74.19%	1.94
Multi+$_{stop,mM}$	186	32	148	32.80%	34.41%	32.79%	138	74.19%	1.94
Multi+$_{stop,Mm}$	186	32	148	32.80%	34.41%	32.79%	138	74.19%	1.94
Multi+$_{stop,MM}$	186	32	148	32.80%	34.41%	32.79%	138	74.19%	1.94
Multi+$_{mm}$	13	0	2	15.38%	0.00%	84.62%	4	30.77%	1.38
Multi+$_{mM}$	13	0	2	15.38%	0.00%	84.62%	4	30.77%	1.38
Multi+$_{Mm}$	13	0	2	15.38%	0.00%	84.62%	4	30.77%	1.38
Multi+$_{MM}$	13	0	2	15.38%	0.00%	84.62%	4	30.77%	1.38
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Precision									
Multi+$_{stop,mm}$	222	222	0	0.00%	0.00%	100.00%	17	7.66%	1.08
Multi+$_{stop,mM}$	222	222	0	0.00%	0.00%	100.00%	17	7.66%	1.08
Multi+$_{stop,Mm}$	22	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	22	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	222	0	0	0.00%	0.00%	100.00%	17	7.66%	1.08
Multi+$_{mM}$	222	0	0	0.00%	0.00%	100.00%	17	7.66%	1.08
Multi+$_{Mm}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Subset Accuracy									
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	-

Table 29: Model analysis of different multi-label classification approaches on the data set GENBASE using the rule-independent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
F-Measure									
Multi+$_{stop,mm}$	47	2	11	14.89%	6.38%	78.73%	7	14.89%	1.15
Multi+$_{stop,mM}$	38	1	4	10.53%	0.00%	89.47%	0	0.00%	1.00
Multi+$_{stop,Mm}$	19	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	19	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	18	0	1	0.00%	5.56%	94.44%	0	0.00%	1.00
Multi+$_{mM}$	18	0	0	0.00%	0.00%	100.00%	1	5.56%	1.06
Multi+$_{Mm}$	14	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	13	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Hamming Accuracy									
Multi+$_{stop,mm}$	194	59	57	0.00%	29.38%	70.62%	104	53.61%	1.89
Multi+$_{stop,mM}$	194	59	57	0.00%	29.38%	70.62%	104	53.61%	1.89
Multi+$_{stop,Mm}$	194	59	57	0.00%	29.38%	70.62%	104	53.61%	1.89
Multi+$_{stop,MM}$	194	59	57	0.00%	29.38%	70.62%	104	53.61%	1.89
Multi+$_{mm}$	144	0	10	1.39%	5.56%	93.05%	58	40.28%	1.69
Multi+$_{mM}$	144	0	10	1.39%	5.56%	93.05%	58	40.28%	1.69
Multi+$_{Mm}$	144	0	10	1.39%	5.56%	93.05%	58	40.28%	1.69
Multi+$_{MM}$	144	0	10	1.39%	5.56%	93.05%	58	40.28%	1.69
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Precision									
Multi+$_{stop,mm}$	137	134	1	0.73%	0.00%	99.27%	60	43.80%	1.58
Multi+$_{stop,mM}$	137	134	1	0.73%	0.00%	99.27%	60	43.80%	1.58
Multi+$_{stop,Mm}$	19	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	19	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	135	0	1	0.74%	0.00%	99.26%	59	43.70%	1.57
Multi+$_{mM}$	135	0	1	0.74%	0.00%	99.26%	59	43.70%	1.57
Multi+$_{Mm}$	13	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{MM}$	13	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{MM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Subset Accuracy									
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	-

Table 30: Model analysis of different multi-label classification approaches on the data set BIRDS using the rule-independent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

Approach	# Rules	# Stopping Rules	# Label Conditions	% Full Label-Dependent	% Partially Label-Dependent	% Not Label-Dependent	# Multi-Label Head Rules	% Multi-Label Head Rules	Avg. # Labels Per Head
	F-Measure								
Multi+$_{stop,mm}$	14	1	0	0.00%	0.00%	100.00%	2	14.29%	1.14
Multi+$_{stop,mM}$	13	1	0	0.00%	0.00%	100.00%	1	7.69%	1.08
Multi+$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	9	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mM}$	8	0	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{Mm}$	6	0	1	16.67%	0.00%	83.33%	0	0.00%	1.00
Multi+$_{MM}$	6	0	1	16.67%	0.00%	83.33%	0	0.00%	1.00
Multi±$_{stop,mm}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{stop,mM}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{stop,Mm}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{stop,MM}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{mm}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{mM}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{Mm}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{MM}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
	Hamming Accuracy								
Multi+$_{stop,mm}$	398	312	1	0.00%	0.25%	99.75%	43	10.80%	1.11
Multi+$_{stop,mM}$	398	312	1	0.00%	0.25%	99.75%	43	10.80%	1.11
Multi+$_{stop,Mm}$	398	312	1	0.00%	0.25%	99.75%	43	10.80%	1.11
Multi+$_{stop,MM}$	398	312	1	0.00%	0.25%	99.75%	43	10.80%	1.11
Multi+$_{mm}$	34	0	0	0.00%	0.00%	100.00%	1	2.94%	1.03
Multi+$_{mM}$	34	0	0	0.00%	0.00%	100.00%	1	2.94%	1.03
Multi+$_{Mm}$	34	0	0	0.00%	0.00%	100.00%	1	2.94%	1.03
Multi+$_{MM}$	34	0	0	0.00%	0.00%	100.00%	1	2.94%	1.03
Multi±$_{stop,mm}$	18	18	0	0.00%	0.00%	100.00%	18	100.00%	6.00
Multi±$_{stop,mM}$	18	0	0	0.00%	0.00%	100.00%	18	100.00%	6.00
Multi±$_{stop,Mm}$	18	0	0	0.00%	0.00%	100.00%	18	100.00%	6.00
Multi±$_{stop,MM}$	18	0	0	0.00%	0.00%	100.00%	18	100.00%	6.00
Multi±$_{mm}$	18	0	0	0.00%	0.00%	100.00%	18	100.00%	6.00
Multi±$_{mM}$	18	0	0	0.00%	0.00%	100.00%	18	100.00%	6.00
Multi±$_{Mm}$	18	0	0	0.00%	0.00%	100.00%	18	100.00%	6.00
Multi±$_{MM}$	18	0	0	0.00%	0.00%	100.00%	18	100.00%	6.00
	Precision								
Multi+$_{stop,mm}$	368	368	0	0.00%	0.00%	100.00%	22	5.98%	1.06
Multi+$_{stop,mM}$	368	368	0	0.00%	0.00%	100.00%	22	5.98%	1.06
Multi+$_{stop,Mm}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{stop,MM}$	6	1	0	0.00%	0.00%	100.00%	0	0.00%	1.00
Multi+$_{mm}$	368	0	0	0.00%	0.00%	100.00%	22	5.98%	1.06
Multi+$_{mM}$	368	0	0	0.00%	0.00%	100.00%	22	5.98%	1.06
Multi+$_{Mm}$	6	0	1	16.67%	0.00%	83.33%	0	0.00%	1.00
Multi+$_{MM}$	6	0	1	16.67%	0.00%	83.33%	0	0.00%	1.00
Multi±$_{stop,mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{stop,Mm}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{stop,MM}$	1	1	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{mm}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{mM}$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Multi±$_{Mm}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
Multi±$_{MM}$	1	0	0	0.00%	0.00%	100.00%	1	100.00%	6.00
	Subset Accuracy								
Multi+$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi+$_{Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{stop,Mm}$	-	-	-	-	-	-	-	-	-
Multi±$_{Mm}$	-	-	-	-	-	-	-	-	-

Table 31: Model analysis of different multi-label classification approaches on the data set SCENE using the rule-independent evaluation strategy. Some approaches did not finish in time. The missing values are indicated by using the label "n/a".

References

Michael Bensimhoun. A note on the mediant inequality. https://commons.wikimedia.org/wiki/File:Extension_of_the_mediant_inequality.pdf, 2013. Accessed August 20, 2016.

Nancy Chinchor. Muc-4 evaluation metrics. In *Proceedings of the Fourth Message Understanding Conference*, pages 22–29, 1992.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45, 2012.

Johannes Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.

Johannes Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2(Mar):721–747, 2002.

Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008.

Valentin Gjorgjioski, Dragi Kocev, and Sašo Džeroski. Comparison of distances for multi-label classification with pcts. In *Proceedings of the Slovenian KDD Conference on Data Mining and Data Warehouses (SiKDD'11)*, 2011.

Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

Thomas Heath. *History of Ancient Greek Mathematics*, volume 1. Clarendon Press, 1921.

Frederik Janssen. *Heuristic Rule Learning*. Dissertation, Knowledge Engineering Group, Technische Universität Darmstadt, Germany, 2012.

Frederik Janssen and Johannes Fürnkranz. On the quest for optimal rule learning heuristics. *Machine Learning*, 78(3):343–379, 2010.

Frederik Janssen and Johannes Fürnkranz. The seco-framework for rule learning. Technical Report TUD-KE-2010-02, Knowledge Engineering Group, Technische Universität Darmstadt, Germany, 2010. URL http://www.ke.tu-darmstadt.de/publications/reports/tud-ke-2010-02.pdf.

Frederik Janssen and Markus Zopf. The seco-framework for rule learning. In *Proceedings of the German Workshop on Lernen, Wissen, Adaptivität - LWA2012*, 2012.

Donald E Knuth. Two notes on notation. *The American Mathematical Monthly*, 99(5):403–422, 1992.

Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S Dhillon. Consistent multilabel classification. In *Advances in Neural Information Processing Systems*, pages 3321–3329, 2015.

Eneldo Loza Mencía. *Efficient Pairwise Multilabel Classification*. Dissertation, Knowledge Engineering Group, Technische Universität Darmstadt, Germany, 2012.

Eneldo Loza Mencía and Frederik Janssen. Learning rules for multi-label classification: a stacking and a separate-and-conquer approach. *Knowledge Engineering Group, Technische Universität Darmstadt, Germany*, 2015. Unpublished work.

Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*, volume 2. Springer, 2005.

D Malerba, G Semeraro, and F Esposito. A multistrategy approach to learning multiple dependent concepts. *Machine learning and statistics: The interface*, pages 87–106, 1997.

Mulan Development Team. Mulan: A java library for multi-label learning – multi-label classification datasets. `http://mulan.sourceforge.net/datasets-mlc.html`, 2016. Accessed November 29, 2016.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Elsevier, 2014.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

Yutaka Sasaki. The truth of the f-measure. *School of Computer Science, University of Manchester*, 2007.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*, 2006.

Shenghuo Zhu, Xiang Ji, Wei Xu, and Yihong Gong. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 274–281. ACM, 2005.