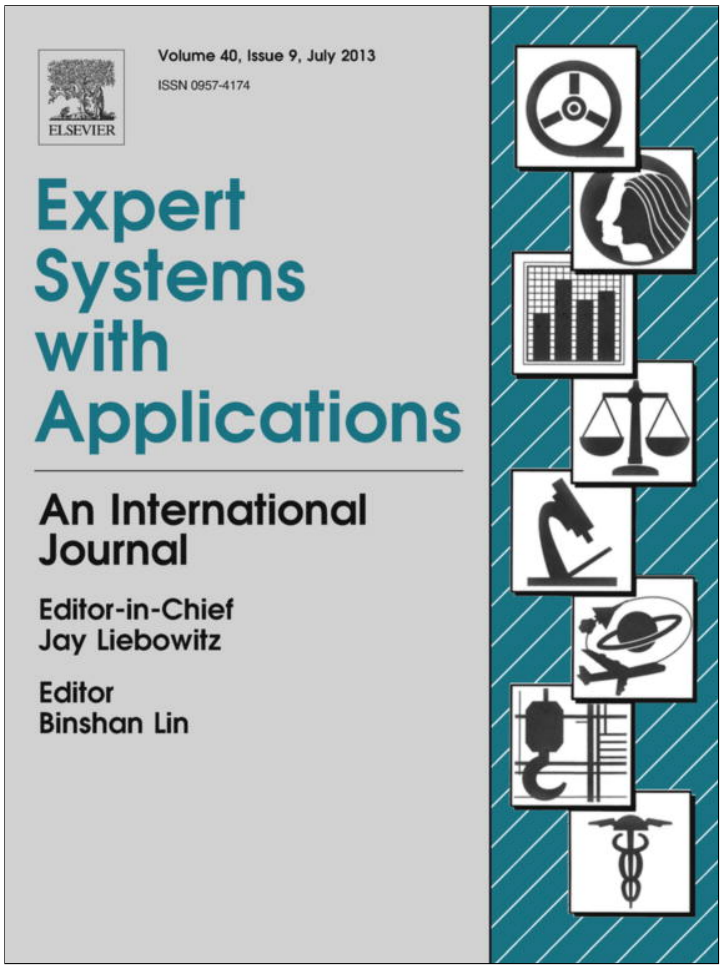

Expert Systems with Applications 40 (2013) 3441–3449

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Variable-constraint classification and quantification of radiology reports under the ACR Index

Stefano Baccianella [1], Andrea Esuli [1], Fabrizio Sebastiani [*,1]

Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy

ARTICLE INFO

Keywords:
Medical reports
Automatic classification
Text classification

ABSTRACT

We apply hierarchical supervised learning technology to the problem of assigning codes from the well-known ACR Index (a "double-hierarchy" classification scheme from the American College of Radiology) to radiology reports. This task is actually two classification tasks in one: the former uses a first hierarchy of codes describing anatomic locations, and the latter uses a second hierarchy of codes describing pathologies, where the two hierarchies are closely intertwined. A requirement of each such classification task is that the document be placed in exactly one node of depth $\geqslant 2$ of the "anatomic location" hierarchy and in exactly one node of depth $\geqslant 3$ of the "pathology" hierarchy; this makes our task a (fairly uncommon) *variable-constraint* classification task, since at the first levels of the hierarchy (2 for anatomic location, 3 for pathology) we need to use a standard "exactly 1 class per document" constraint, while at the lower levels we need to use an "at most 1 class per document" constraint. We have used a large dataset of about 250,000 radiology reports written in Italian and an adaptation of our TREEBOOST.MH learning algorithm to variable-constraint classification. Notwithstanding the extreme difficulty of the task (given by the fact that the two codes had to be picked out of a pool of 719 codes for anatomic location and 5269 codes for pathology, respectively) our system displayed good accuracy, indicating that it may represent a viable tool for semi-automated classification of medical reports. We also analyzed the *quantification* accuracy of our system (i.e., the ability of the system at correctly estimating the frequency of the individual codes), a concern of special interest in epidemiology; the results show that our system has excellent quantification accuracy, making this system a valuable tool for the fully automated coding of radiology reports for epidemiological purposes.

1. Introduction

Classifying (also known as "coding") medical reports is a standard practice in hospitals and health-related organizations, since it helps in organizing the work of medical personnel, aids communication among different departments of the same organization, helps the work of the hospital administration, generates data for epidemiological studies, and improves the service to the patient by giving a clearer organization to the patient's medical documentation.

However, coding medical reports is a hard task for doctors, since (i) the classification schemes are large and difficult to remember, (ii) doctors always work in critical time conditions, and coding requires time to be performed correctly, and (iii) doctors' real expertise is in curing patients, and not in classifying data.

As a result, it would be useful for doctors to have an automatic coding system that can either (a) recommend the correct codes to use for a given report, or (b) suggest the correct codes for reports that it deems likely to be incorrectly coded, or (c) automatically code entire batches of legacy, yet uncoded reports.

In this work we experiment on the automatic coding of radiology reports based on the ACR Index, a standard classification scheme for the radiology discipline; we are not aware of previous attempts at doing automated coding against the ACR Index. Given that ACR is in the form of a taxonomy, we adopt a hierarchical learning approach; this allows us to perform learning and coding efficiently despite the huge size of the coding scheme. This coding task is actually two (almost independent) coding tasks in one, the former consisting of the assignment of a code from a first hierarchy of codes describing anatomic locations, and the latter consisting of the assignment of a code from a second hierarchy describing pathologies, where the two hierarchies are closely intertwined. A requirement of each such classification task (see Section 2 for details) is that the document be placed in exactly one node of depth $\geqslant 2$ of the "anatomic location" hierarchy and in exactly one node of depth $\geqslant 3$ of the "pathology" hierarchy. This requirement on the

* Corresponding author. Tel.: +39 050 3152892; fax: +39 050 3153464.
E-mail addresses: Stefano.Baccianella@isti.cnr.it (S. Baccianella), Andrea.Esuli@isti.cnr.it (A. Esuli), fabrizio.sebastiani@isti.cnr.it (F. Sebastiani).

[1] The order in which the authors are listed is purely alphabetical; each author has given an equally important contribution to this work.

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.12.052

depth of the nodes makes our task a (fairly uncommon) *variable-constraint* classification task, since at the first levels of the hierarchy (2 levels for anatomic location, 3 levels for pathology) we need to use a standard "exactly 1 class per document" constraint, while at the lower levels we need to use an "at most 1 class per document" constraint.

We have run our experiments, using an adaptation to variable-constraint classification of our TREEBOOST.MH hierarchical algorithm, on a large dataset consisting of about 250,000 radiology reports written in Italian by medical personnel of Policlinico Umberto I, one of the largest hospitals in Rome. Note that coding radiology reports written in Italian has additional difficulties with respect to coding analogous reports written in English, since there is a wealth of language tools and resources available for English that is not available for Italian. For instance, the RadLex standard lexicon of radiology terms developed by the Radiological Society of North America (RSNA) (Langlotz, 2006) is available for English but not for Italian, and this certainly makes the task of producing highly accurate classifiers of radiology reports harder. This paper may thus be seen as testing methods for the classification of medical reports written in resource-poor languages.

We have analyzed the results of our experiments not just in the (pretty standard context) of classification, but also according to the novel framework of *quantification*, a task concerned not with deciding whether an individual yet uncoded report should be attributed a given code or not, but with correctly estimating the percentage of yet uncoded reports that should be attributed the code. As we will argue, quantification has important applications that make it worthwhile studying it *per se*.

The paper is organized as follows. In Section 2 we describe the ACR classification hierarchy and structure. In Section 3 we present the methods and algorithms we have used to tackle the problem, while in Section 4 we report the results of our classification experiments. Section 5 reanalyzes the results of these experiments under the light of quantification. In Section 6 we discuss related works from the literature, and we conclude in Section 7.

2. The ACR classification scheme

The American College of Radiology's Index for Radiological Diagnoses (known as the ACR Index – see American College of Radiology, 1992) is a classification scheme developed by the American College of Radiology[2] aimed at allowing the categorization of radiology-based documentation, including radiology reports.

The ACR Index is actually two classification schemes in one, since it caters for (i) classification according to the *anatomic location* that was the subject of investigation (i.e., the part of the body where the radiologic image has been taken), and (ii) classification according to the *pathology* suspected or detected by the radiologist. Both classification schemes are hierarchical in nature, and nodes at increasing levels of depth in either hierarchy identify anatomic locations/ pathologies at increasing levels of detail (e.g., the higher-level nodes of the "anatomic location" hierarchy identify macro-components of the body, such as Breast, while lower-level nodes identify smaller components, such as Nipple).

By convention, an ACR code is a pair of digit sequences separated by a dot (e.g., `05.311`). The numeric code for the anatomic location identifier appears to the left of the dot, while the numeric code for the pathology appears to the right of the dot. Both numeric codes have at least two digits, with the leftmost digits in each code representing more general information (i.e., concepts high up in the hierarchy) and the rightmost digits in each code representing more specific information (i.e., concepts at the lower levels of the hierarchy). For instance, ACR code `05.311` identifies documents about the breast (identified by the leading `0`) and in particular about the nipple (`05`), where the radiologist either has found or suspects the presence of a benign neoplasm (`.31`) of the fibroadenoma type (`.311`).

Both the anatomic location hierarchy and the pathology hierarchy have 10 first-level nodes (with the root conventionally taken to be at level – or *depth* – 0), also known as "macro-areas", which are the same for both hierarchies; they are Breast, Skull, Face, Spine, SkeletalSystem, Heart, Lung, GastrointestinalSystem, GenitourinarySystem, VascularSystem, LymphaticSystem. For the rest, the two hierarchies are different: for each macro-area, the anatomic location hierarchy defines a specific sub-hierarchy of anatomic sub-locations, and the pathology hierarchy defines a specific sub-hierarchy of pathologies that may affect that macro-area. For a given macro-area, the anatomic location and pathology sub-hierarchies differ in structure. Note that the first digit of the anatomic location code is "implicitly" present as a prefix to the pathology code; that is, code `05.311` should actually be understood as `05.0311`, since there are actually 10 different pathology hierarchies, one for each anatomic location macro-area.

The anatomic location hierarchy has a maximum of 5 levels (including the root). Each level has a maximum of 10 nodes, and the leaves are all located from level 2 to level 4. The hierarchy has a total of 730 nodes (including the root), 641 of which are leaves. The pathology identifier hierarchy instead has a maximum of 7 levels (including the root). Each level has a maximum of 10 nodes,[3] and the leaves are located from level 4 to level 7. The hierarchy has a total of 5,380 nodes (including the root), 4,404 of which are leaves.

Classification according to the ACR Index consists (see American College of Radiology, 1992) in assigning to the report exactly one code of depth $\geqslant 2$ (i.e., just assigning a macro-area is not allowed) from the anatomic location hierarchy *and* exactly one code of depth $\geqslant 3$ from the pathology hierarchy. As a result, automatic classification according to the ACR Index is an extremely hard task, since the two classification tasks consist in:

1. Identifying *the* right anatomic location class of depth $\geqslant 2$ from a pool of (730–11) = 719 legally assignable classes (11 codes are subtracted since 1 root and 10 first-level nodes are not assignable); and
2. identifying *the* right pathology class of depth $\geqslant 3$ from a pool of (5,380–111) = 5,269 legally assignable classes (111 codes are subtracted since 1 root, 10 first-level nodes and 100 second-level nodes are not assignable).

This is a bit like finding a needle in haystack.

3. Attributing ACR codes to radiology reports using hierarchical classification

The system we have tested is an adaptive system for automatically coding radiology reports under any user-specified classification scheme; given such a classification scheme, the system automatically generates an automatic classifier for this classification scheme. Our system is based on supervised machine learning technology, according to which the system learns from manually coded radiology reports the characteristics that a new radiology report should have in order to be attributed the code. The manually coded radiology reports that are fed to the system for the purpose

[2] http://www.acr.org/

[3] The fact that in both hierarchies each level has a maximum of 10 nodes is due to the fact that, when the ACR Index was originally defined, it was decided that each node in the hierarchy was to be represented via a numeric sequence of decimal digits, and this obviously limits the number of nodes representable by a given digit to 10.

of generating the binary classifiers are called the *training reports*. The training reports need to include *positive examples* of the code (i.e., radiology reports to which a human classifier has attributed the code) and *negative examples* of the code (i.e., radiology reports to which a human classifier has decided not to attribute the code). By examining both, the system identifies the discriminating characteristics of the radiology reports, i.e., the characteristics that will help the binary classifier in deciding whether to attribute a given code or not to a yet uncoded radiology report.

3.1. The TREEBOOST.MH learning algorithm

As the learning method we have used a modified version of the TREEBOOST.MH algorithm (Esuli, Fagni, & Sebastiani, 2008), a learning algorithm explicitly devised for generating classifiers for hierarchically organized classification schemes.

Before describing our learning method, let us first define our notation and terminology. Given a set of textual documents D and a predefined class (also known as *label*, or *code*) c_j, a *binary classifier* for c_j is a function $\widehat{\Phi}^j : D \rightarrow \mathbb{R}$ such that the sign of $\widehat{\Phi}^j(d_i)$, indicated by $sgn(\widehat{\Phi}^j(d_i))$, is interpreted as the classifier's prediction whether d_i belongs to c_j or not, and the absolute value of $\widehat{\Phi}^j(d_i)$, indicated by $|\widehat{\Phi}^j(d_i)|$, is interpreted as the confidence that the classifier has in this prediction, with higher values indicating higher confidence. A *multi-label classifier* for a set of classes C is a set $\{\widehat{\Phi}^j : D \rightarrow \mathbb{R}\}_{c_j \in C}$ of binary classifiers, where each $\widehat{\Phi}^j$ decides whether a document belongs to c_j or not independently of the classifiers for the other classes in C; as a consequence, a multi-label classifier can attribute to a document d_i zero, one, or several classes in C at the same time.

Let $C = \langle I, L \rangle$ be a tree-structured set of classes, where $I = \{i_1, \ldots, i_m\} \subseteq C$ is the set of *internal* (i.e., nonleaf) *classes* of C and $L = \{l_1, \ldots, l_n\}$ is the set of *leaf classes* of C. By r we denote the root class of C; note that $r \in L$ if the tree is degenerate, i.e., it consists of the root only, while $r \in I$ otherwise. In our notation, we thus have $C = \{c_1, \ldots, c_{m+n}\} = r \cup \{i_1, \ldots, i_m\} \cup \{l_1, \ldots, l_n\}$. For each class $c_j \in C$ we will use the notation $\downarrow(c_j)$ to indicate the set of children classes of c_j.

The TREEBOOST.MH algorithm was originally devised for generating multi-label (hierarchical) classifiers, i.e., classifiers that can associate to a document d_i zero, one, or several classes in C at the same time. In TREEBOOST.MH the multi-label hierarchical classification problem is broken down into several smaller multi-label "flat" (i.e., non-hierarchical) classification problems, one for every internal class i_s of the hierarchy. For each $i_s \in I$ the algorithm calls a flat learning algorithm that generates a set $\{\widehat{\Phi}^j\}_{c_j \in \downarrow(i_s)}$ of binary classifiers, one for each child c_j of i_s, and carries on recursively until a binary classifier is generated for each nonroot node of C.

When classifying a new document d_i, classification is then performed in "Pachinko machine" style (Koller & Sahami, 1997): the test document is first submitted to the classifiers corresponding to the top-level nodes, and recursively percolates down to the lower levels of the hierarchy only if the classifiers at the higher levels have deemed that the document belong to their associated class. In this way, entire subtrees are pruned from consideration, which allows exponential savings at classification time (Chakrabarti, Dom, Agrawal, & Raghavan, 1998; Koller & Sahami, 1997).

3.2. Modifying TREEBOOST.MH to account for the semantics of the ACR hierarchy

The multi-label nature of TREEBOOST.MH (and all other hierarchical learning algorithms, for that matter) is slightly at odds with the constraints inherent in the ACR hierarchy, that (as noticed in Section 2) prescribe that both numeric codes (the one for anatomic location and the one for pathology) have at least two digits; i.e., for each document d_i

- exactly one node c_j of depth 1 in the anatomic location hierarchy (corresponding to the anatomic "macro-area") must be selected, and
- at least one node in $\downarrow(c_j)$ in the anatomic location hierarchy *and* at least one node in $\downarrow(\downarrow(c_j))$ in the pathology hierarchy (the former node having thus depth 2 and the latter node having depth 3) must be selected.

This means that, for both taxonomies:

1. At the levels of depth $\leqslant 2$ (anatomic location) and $\leqslant 3$ (pathology) we have an "exactly-1" constraint (i.e., at both levels each document must be assigned to exactly one class). For solving this problem, both at the first 2 levels of the anatomic location hierarchy and at the first 3 of the pathology hierarchy we classify d_i in the class
$$\arg\max_{c_j} \widehat{\Phi}^j(d_i)$$
i.e., in the class which receives the highest score for d_i.
2. At the levels of higher depth we have instead an "at-most-1" constraint (i.e., at each such levels each document may be assigned to 0 or 1 classes): if d_i has been already assigned to class c_j in a previous step, then we assign d_i to the class
$$\arg\max_{c_j} \widehat{\Phi}^j(d_i)$$
if d_i has received a score higher than 0 for this class; otherwise no class is assigned to d_i at this level.

This modification effectively turns TREEBOOST.MH into what we might call a *variable-constraint hierarchical learning algorithm*.

3.3. Generating flat multi-label classifiers via MP-BOOST

In order to generate a flat multi-label classifier at each recursive step of the TREEBOOST.MH algorithm (see Section 3.1), we use the MP-BOOST "boosting" learning algorithm (Esuli, Fagni, & Sebastiani, 2006). Boosting algorithms have strong justifications from computational learning theory (Meir & Rätsch, 2003) and, at the same time, are among the supervised learning algorithms that have obtained the best performance in several learning tasks. MP-BOOST is a variant of ADABOOST.MH (Schapire & Singer, 2000), and has been shown in Esuli et al. (2006) to obtain considerable effectiveness improvements with respect to ADABOOST.MH.

MP-BOOST works by iteratively generating, for each class c_j, a sequence $\widehat{\Phi}^j_1, \ldots, \widehat{\Phi}^j_S$ of classifiers (called *weak hypotheses*). A weak hypothesis is a function $\widehat{\Phi}^j_s : D \rightarrow \mathbb{R}$ where (as in Section 3.1) $sgn\left(\widehat{\Phi}^j_s(d_i)\right)$ represents the prediction of $\widehat{\Phi}^j_s$ on whether d_i belongs to c_j and $|\widehat{\Phi}^j_s(d_i)|$ represents the confidence that $\widehat{\Phi}^j_s$ has in this decision.

At each iteration s MP-BOOST tests the effectiveness of the most recently generated weak hypothesis $\widehat{\Phi}^j_s$ on the training set, and uses the results to update a distribution D^j_s of weights on the training examples. The initial distribution D^j_1 is uniform by default. At each iteration s all the weights $D^j_s(d_i)$ are updated, yielding $D^j_{s+1}(d_i)$, so that the weight assigned to an example correctly (resp., incorrectly) classified by $\widehat{\Phi}^j_s$ is decreased (resp., increased). The weight $D^j_{s+1}(d_i)$ is thus meant to capture how ineffective $\widehat{\Phi}^j_1, \ldots, \widehat{\Phi}^j_s$ have been in guessing whether training document d_i belongs to class c_j or not. By using this distribution, MP-BOOST generates a new weak hypothesis $\widehat{\Phi}^j_{s+1}$ that concentrates on the examples with the highest weights, i.e., those that had proven harder to classify for the previous weak hypotheses.

The overall prediction on whether d_i belongs to c_j is obtained as a sum $\widehat{\Phi}^j(d_i) = \sum_{s=1}^{S} \widehat{\Phi}^j_s(d_i)$ of the predictions made by the weak hypotheses. The final classifier $\widehat{\Phi}^j$ is thus a *committee* of S classifiers, a committee whose S members each cast a weighted vote (the vote being the binary decision $sgn\left(\widehat{\Phi}^j_s(d_i)\right)$, the weight being the confidence $\left|\widehat{\Phi}^j_s(d_i)\right|$) on whether d_i belongs to c_j. For the final classifier $\widehat{\Phi}^j$ too, $sgn\left(\widehat{\Phi}^j(d_i)\right)$ represents the binary decision as to whether d_i belongs to c_j, while $|\widehat{\Phi}^j(d_i)|$ represents the confidence in this decision.

3.4. Sets of features

Like all learning algorithms, TREEBOOST.MH needs each of our radiology reports to be represented in vectorial form. To this end, in all the experiments discussed in this paper stop words have been removed, punctuation has been removed, all letters have been converted to lowercase, numbers have been removed, and stemming has been performed by means of Porter's stemmer. Word stems are thus our indexing units. Since MP-BOOST, which is recursively called by TREEBOOST.MH, requires binary input, only the presence/ absence of these word stems in the document is recorded, and no weighting is performed.

4. Experiments

4.1. Experimental setting

4.1.1. Dataset

The dataset we have used in this work (hereafter called the *UmbertoI* dataset) consists of a set of 248,583 free-text radiology reports written (in Italian) by medical personnel of the Istituto di Radiologia of Policlinico Umberto I, one of the largest hospitals in Rome. Consistently with the semantics of the ACR classification scheme that we have specified in Section 2, all the reports are associated to one and only one anatomic location code, and to one and only one pathology code.

The raw dataset, as we received it from the Policlinico Umberto I personnel, contained 132 reports with an invalid ACR code. The reason why the ACR code was invalid in these reports was that either the anatomic location identifier, or the pathology identifier, or both, consisted of 1 digit only (plus the first digit of the anatomic location code that is "implicitly" present as a prefix to the pathology code – see Section 2), and an ACR code needs at least two digits for each identifier (plus the implicit digit above). We have applied the following correction routine, consisting of three simple rules, and reinstated the 132 reports in the dataset:

1. If the anatomic location identifier consists of only one digit, then append a 0 to the right of it, since 0 as second digit means "Generic".
2. If the pathology identifier is only one digit long and is a 1, then append 1 to the right of it, since the first 1 in the pathology identifier means "Treatment" and a second 1 means "Generic treatment".
3. If the pathology identifier is only one digit long and is not a 1, then append a 9 to the right of it, because a 9 as a second digit means "No further specification".

In addition we discarded all the reports consisting of 4 words or less (this resulted in discarding a total of 4,519 reports), since such reports typically consist of uninformative keyphrases such as `Immagini in visione` ("Images still under examination"), `Esame non eseguito` ("Examination not performed"), etc. Such reports cannot obviously be classified based on the text of the report alone.

In Table 1 we report some statistics about the UmbertoI dataset. For each macro-area the table indicates the number of distinct ACR codes (i.e., of distinct combinations of anatomic location code and pathology code), the number of distinct anatomic location codes, the number of distinct pathology codes, the average length of the reports (in number of words) and the number of reports. Note that only 431 anatomic location codes and 1,198 pathology codes are represented in this dataset, fewer than the 719 anatomic location codes and 5,269 pathology codes that can legally be assigned according to the ACR rules; this means that many ACR Index codes were never used by the radiologists who manually annotated this dataset. Note also that the dataset is highly unbalanced, since 57.3% of the documents belong to three macro-areas altogether (Lung, GastrointestinalSystem, and Breast). This situation is very similar to what we are confronted with in most text classification applications (see e.g., Hersh, Buckley, Leone, & Hickman (1994), Lewis, Yang, Rose, & Li (2004), Liu et al. (2005)), where class frequency (i.e., the number of positive examples per class) is often described by a power law, in which very few classes have many positive examples and are followed by a long tail of many classes with very few, or sometimes no, positive examples (see also Table 3 on this). In our case, this is obviously due to the fact that not all pathologies catered for by the ACR classification scheme occur with the same frequency, and not all anatomic locations catered for by the same scheme are affected with the same frequency.

4.1.2. Evaluation measure

As a measure of effectiveness that combines the contributions of *precision* (π) and *recall* (ρ) we have used the well-known F_1 function, defined as

$$F_1 = \frac{2\pi\rho}{\pi+\rho} = \frac{2TP}{2TP+FP+FN} \quad (1)$$

and which corresponds to the harmonic mean of precision and recall, where TP stands for the number of true positives, FP for the number of false positives, and FN for the number of false negatives. Note that F_1 is undefined when $TP = FP = FN = 0$; in this case, consistently with most other works in the literature, we take F_1 to equal 1, since the classifier has correctly classified all documents (as negative examples).

We compute both *microaveraged* F_1 (denoted by F_1^{μ}) and *macroaveraged* F_1 $\left(F_1^{M}\right)$. F_1^{μ} is obtained by (i) computing the class-specific values TP_i, (ii) obtaining TP as the sum of the TP_i's (same for FP and FN), and then (iii) applying the $F_1 = \frac{2TP}{2TP+FP+FN}$ formula. F_1^{M} is obtained by first computing the class-specific F_1 values and then averaging them across the various classes in the classification scheme. In principle, F_1^{μ} and F_1^{M} may return very different results for the same experiment, since the former tends to reward classi-

Table 1
Statistics for the UmbertoI dataset.

Macro-Area	# Codes	# Anat. location codes	# Pathology codes	Avg length of doc	# Docs
Breast	281	10	90	80	45,237
Skull	165	38	70	106	3,218
Face	427	44	121	148	17,558
Spine	206	33	66	117	22,073
SkeletalSystem	511	54	118	131	26,535
Heart	330	42	108	197	5,947
Lung	361	20	133	72	47,919
GastrointestinalSystem	1,059	66	172	179	46,851
GenitourinarySystem	849	53	205	194	17,804
VascularSystem	511	71	115	165	10,922
Total	4,700	431	1,198	139	244,064

fiers that behave well on *frequent classes* (i.e. classes with many positive examples), while classifiers that perform well also on infrequent classes are emphasized by the latter.

4.1.3. Experimental protocol

As discussed in Section 2, ACR codes have two components: the numeric code before the dot identifies the anatomic location, while the numeric code after the dot identifies the pathology. Starting from this observation we tackle the problem of assigning the ACR code to a radiology report by splitting it into two hierarchical classification problems: the first problem is the assignment of the anatomic location code, while the second is the assignment of the pathology code. As discussed in Section 2, the two hierarchies have the same first-level codes, identifying the anatomic macro-areas. As a result the two classifiers are trained with the same first-level training sets, and this results in generating exactly the same classifiers for the level 0 of the hierarchy.

For our experiments we have randomly split the *UmbertoI* dataset into a training set, containing $\frac{1}{3}$ of the reports of the entire dataset (for a total of 82,861 documents), and a test set, consisting of the other $\frac{2}{3}$ (165,722 documents); this is a challenging split, since a low number of training documents makes the classification task harder, while a high number of test documents makes the results of the experiments more credible. Each class has a minimum of 1 and a maximum of 16,477 training examples.

In a supervised machine learning setting, only classes for which there is at least one positive training example can be dealt with; for the classes for which no positive training example is available, it is obvious that no classifier can be generated. As a result, in our experiments we had to remove from consideration 68 of the 431 anatomic location codes and 368 of the 1,198 pathology codes present in the dataset, leaving us with 363 anatomic location codes and 830 pathology codes to work with (see Table 2).

4.2. Results

The results of our experiments are displayed in Table 3, where we report F_1 results (which are computed as averages across the 363 anatomic location and 830 nodes for which we have generated classifiers) broken down by groups of classes that have a number of training examples in a specified range. For instance, the 1st row of the table reports the average (both F_1^μ and F_1^M) accuracy values for the 242 anatomic location codes that have less than 20 training examples, and the average accuracy values for the 675 pathology codes that have less than 20 training examples.

Table 2
Numbers of codes in the ACR Index and in the UmbertoI dataset.

	Anatomic location	Pathology
# Codes in the ACR Index	730	5,380
# Codes legally assignable	719	5,269
# Codes assigned in the UmbertoI dataset	431	1,198
# Codes assigned in the UmbertoI training set	363	830

Table 3
Average F_1^μ and average F_1^M results for nodes whose number of training examples is in a prespecified range.

# Examples	Anatomic location			Pathology		
	# Codes	F_1^μ	F_1^M	# Codes	F_1^μ	F_1^M
1–20	242	0.201	0.223	675	0.102	0.262
21–50	45	0.200	0.206	48	0.208	0.240
51–200	46	0.304	0.319	62	0.228	0.250
201–500	14	0.399	0.380	17	0.311	0.299
501–1000	4	0.421	0.450	11	0.323	0.300
1001–5000	7	0.471	0.479	14	0.510	0.465
5001–10000	3	0.629	0.633	2	0.599	0.575
10001+	2	0.853	0.833	1	0.827	0.827
All	363	0.611	0.254	830	0.490	0.266

Table 4
Average F_1^μ and F_1^M results for nodes at different depths in the hierarchy.

Depth	Anatomic location			Pathology identifier		
	# Codes	F_1^μ	F_1^M	# Codes	F_1^μ	F_1^M
1	10	0.767	0.682	10	0.767	0.682
2	97	0.628	0.291	88	0.521	0.293
3	323	0.388	0.250	414	0.495	0.250
4	15	0.257	0.103	613	0.526	0.275
5	–	–	–	175	0.274	0.298
6	–	–	–	6	0.471	0.583

Table 4 reports instead the accuracy as computed at a certain depth in a given taxonomy. For instance, the 2nd row of the table reports the average (both F_1^μ and F_1^M) accuracy values for the 97 anatomic location codes of depth 2 and for the 88 pathology codes of depth 2. In the same row, for each document that has been classified at a level deeper than 2, only the correctness of its assignment at level 2 is considered; for instance, wrongly attributing anatomic location code `031` to a document instead of attributing it the correct code `033` counts, at this level, as a *correct* classification, since the first two digits of the code have been correctly identified.

The first insight that can be gained by looking at Table 3 is that the accuracy of classification is, as could be expected, strongly dependent on the number of training examples for the class, with more frequent classes obviously obtaining higher accuracy. This is also confirmed by the results in Table 4, which shows that classes at the higher levels of the hierarchy are the ones on which higher accuracy is obtained. In fact, the classes at the higher levels of the hierarchy are obviously the most frequent, since whenever a document d_i is a training document for class c_j it is *a fortiori* also a training example for any ancestor class of c_j (e.g., a training example for anatomic area Breast (code `0`) is also a training example for anatomic area Nipple (code `05`)). As a consequence, the codes with the highest number of training examples tend to be the top-level ones.

If we consider all codes, irrespectively of the number of training examples, the average accuracy of our system (see last row of Table 3) is 0.611 (anatomic location) and 0.490 (pathology) in terms of microaveraged F_1. The second observation is thus that, relatively to the *a priori* difficulty of the task, the results are reasonably good, since we are talking of a classification task in which *the* correct class must be picked from a large number of legally assignable classes (719 for anatomic location and 5,269 for pathology), which pretty much amounts to finding a needle in a haystack (for instance, in the classification-by-pathology case the *a priori* probability of picking the right class is 1/5269 = 0.00018). For a machine learning algorithm, the task is also made difficult by the presence within the training set of several inconsistently coded duplicate reports, which obviously confuse the learning algorithm (and would likewise confuse any human coder who attempted to learn coding reports by examining these training examples).

Are $F_1^\mu = 0.611$ (anatomic location) and $F_1^\mu = .490$ (pathology) "good enough" for allowing human coders of radiology reports to be entirely replaced by a coding system such as the one we have presented?

Scientifically speaking, a correct answer to this question would require a thorough inter-coder agreement study, where the agreement (i.e., relative accuracy) between two different human radiol-

ogists R_1 and R_2 at picking the correct codes for the reports in the test set is compared with the analogous agreement between either of R_1 and R_2 and our system. In this respect, F_1 is an ideal function for measuring accuracy since it is symmetric, i.e., it is invariant with respect to swapping the predicted codes and the true codes. One might thus compare $F_1(R,R_2)$, $F_1(System,R_1)$ and $F_1(System,R_2)$, thanks to the fact that F_1 does not require specifying who of the two human coders plays the role of the "gold standard." Claiming that the system performs better than humans at coding would thus entail showing that $avg(F_1(System,R_1),\ F_1(System,R_2)) > F_1(R,R_2)$. Unfortunately, it was impossible for us to perform such a study, since none of the reports in the UmbertoI dataset had independently been coded by two human coders.

In the absence of a thorough inter-coder agreement study, it seems clear that these results are not good enough justify the adoption of a fully-automated coding system to replace human coders *tout court*. However, while further research on coding radiology reports is certainly needed, we think that our system could already be used in partially automated contexts, such as in recommending the correct codes to use for a given report, or in suggesting the correct codes for reports that the system deems likely to be incorrectly coded.

In terms of coding *efficiency* (i.e., time taken to code the data in the test set), we note that the coding of the 165,722 test documents required only 1 h and 29 min (which also include the time taken to generate the internal representations of the documents from the raw text) on a standard 10-core, 3 GHz machine with 6 GB RAM, which means coding at a speed of more than 31 reports per second. If we consider that each document was coded against two large hierarchies (consisting of 719 and 5,269 codes, respectively), this speaks of a very good efficiency.

5. Switching from classification to quantification

In this section we look at our classification experiments under a different angle, that of *quantification via classification* (Esuli & Sebastiani, 2010; Forman, 2008). Let us formalize this.

As already introduced in Section 3.1, *classification* for class c_j may be defined as the task of generating a binary classifier $\widehat{\Phi}^j : D \to \mathbb{R}$ such that, "for as many $d_i \in D$ as possible", $sgn(\widehat{\Phi}^j(d_i)) = \Phi^j(d_i)$, where Φ^j is our "ground truth".[4] In other words, a good classifier must correctly classify as many individual (test) documents as possible.

Given a classification function $\Phi^j : D \to \mathbb{R}$ let us define $freq(\Phi^j,D)$ as the (relative) *frequency* of Φ^j in D, i.e., as the fraction (or percentage) of the items $d_i \in D$ such that $\Phi^j(d_i) = 1$. *Quantification* (via classification) for class c_j may now be defined as the task of generating a binary classifier $\widehat{\Phi}^j : D \to \mathbb{R}$ such that $freq(\widehat{\Phi}^j,D)$ is "as close as possible" to $freq(\Phi^j,D)$. In other words, a classifier $\widehat{\Phi}^j$ is good at quantification (or: "is a good quantifier") if it estimates as accurately as possible the percentage $freq(\Phi^j,D)$ of documents that actually belong to a class c_j.

On the surface it would seem that the more we improve the accuracy of classification, the more we improve the accuracy of quantification, and that the only way to improve the ability of a classifier to correctly estimate the distribution of test documents across classes is to improve its ability at classifying individual documents. Unfortunately, we contend this is not necessarily true. To see this, one only needs to look at the definition of F_1: it is evident from Eq. (1) that F_1 deteriorates with $(FP + FN)$, and not with $|FP - FN|$, as would instead be required of a function that truly optimizes quantification. For example, according to F_1 a classifier $\widehat{\Phi}_1$ for which $FP = 50$ and $FN = 50$ is worse (all other things being equal) than a classifier $\widehat{\Phi}_2$ for which $FP = 0$ and $FN = 10$. However, $\widehat{\Phi}_1$ is better than $\widehat{\Phi}_2$ according to any reasonable measure for evaluating quantification accuracy; indeed, $\widehat{\Phi}_1$ is a perfect quantifier, since FP and FN are equal and thus compensate each other, so that the distribution of the test items is estimated perfectly.

Quantification is still a fairly unexplored task, having drawn the attention of researchers only in recent years (Bella, Ferri, Hernández-Orallo, & Ramírez-Quintana, 2010; Esuli & Sebastiani, 2010, 2010; Forman, 2008; Xue & Weiss, 2009). Its applicative interest derives from the fact that in several applications, estimating the accuracy of quantification is more interesting than estimating the accuracy of classification. For instance, in a market research application in which a questionnaire asks about the respondent's perception of a given ad campaign, who administers the questionnaire is likely *not* interested in whether John Smith's textual answer belongs or not to the class "Liked the campaign", but is instead likely interested in knowing the *percentage* of responses that belong to the class. Similarly, given a large set of star-rated reviews of a given MP3 player, a survey specialist is likely *not* interested in the fact that it has been rated "4 stars" by John Smith, but is likely interested in the percentage of reviewers that have rated it "4 stars".

In the medical domain, quantification is interesting for epidemiological studies. Here, a researcher may be interested in the percentage of patients that have been diagnosed with a benign neoplasm (`.31`); or in the percentage of patients for which the diagnosed benign neoplasm (`.31`) is of the fibroadenoma type (`.311`); or may be interested in monitoring how the percentage of patients that have been diagnosed with a benign neoplasm evolves with time.

As a consequence, we have tested the quantification accuracy of our system on the UmbertoI dataset; this essentially means taking the results of our train-and-test run described in Section 4 and evaluating them according to a measure of quantification accuracy. For this we have chosen the simple *percentage discrepancy* (*PD*) measure used in Esuli and Sebastiani (2010), and defined as

$$PD(\widehat{\Phi}^j, D) = |freq(\Phi^j, D) - freq(\widehat{\Phi}^j, D)|$$

i.e., the absolute value of the difference between the true and the predicted frequency of the class; low values are better, and the perfect quantifier has $PD(\widehat{\Phi}^j, D) = 0$. For example, if the predicted frequency of class c_j is 0.34 and its true frequency is 0.36, then $PD = |0.34 - 0.36| = 0.02$. For better clarity we will hereafter express *PD* values as percentages instead of as fractions, e.g., writing $PD = 2\%$ instead of $PD = 0.02$.

Table 5 reports *PD* results broken down by groups of classes that have a number of training examples in a specified range. We report both the average and the maximum of the values of *PD* across all

Table 5
Average *PD* and maximum *PD* results for nodes whose number of training examples is in a prespecified range. Lower values are better, best is 0%, worst is 100%.

# Examples	Anatomic location			Pathology		
	# Codes	avg (*PD*) (%)	max (*PD*) (%)	# Codes	avg (*PD*) (%)	max (*PD*) (%)
1–20	242	0.00	0.12	675	0.01	0.91
21–50	45	0.02	0.43	48	0.01	0.18
51–200	46	0.05	0.87	62	0.05	0.53
201–500	14	0.11	0.31	17	0.12	0.64
501–1000	4	0.53	1.54	11	0.22	1.12
1001–5000	7	0.26	0.75	14	0.43	1.53
5001–10000	3	1.60	2.52	2	0.60	0.92
10001+	2	1.60	2.42	1	0.40	0.44
All	363	0.05	2.52	830	0.02	1.53

[4] Consistently with most mathematical literature we use the caret symbol (^) to indicate estimation.

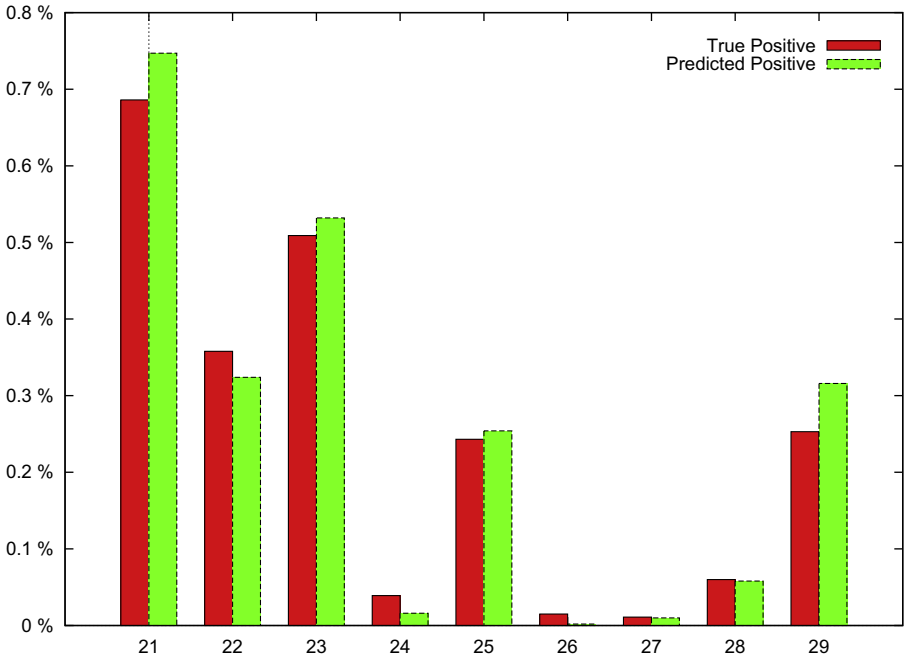

Fig. 1. True percentages (first bar) against predicted percentages (second bar) for the ten pathology codes children of code 2, corresponding to macro-area Skull; code 20 is not displayed since both the true and the predicted percentage were 0. The worst value of *PD* across these ten codes (0.06%) is obtained for code 29, where 0.25% is the true percentage and 0.31% is the percentage predicted by our system. The average *PD* across these ten codes is 0.02%.

the classes in the group (the maximum value represents of course a worst-case scenario). The results show that our system is extremely good at quantification. Across the 363 anatomic location codes of our experiment, the average value of *PD* is 0.05%, an extremely low value, while it is even lower (0.02%) across the 830 pathology codes; as an example, *PD* = 0.05% is the value one would obtain by predicting a given class frequency as 30.05% (or 29.95%) while the true class frequency is 30%. The values of *PD* are never higher than 2.53% for anatomic location and 1.53% for pathology, which indicates that class frequencies are overestimated or underestimated at most marginally.

In order to better visualize this, in Fig. 1 we display the quantification accuracy for the ten pathology codes children of code 2 (corresponding to macro-area Skull) in histogram form. For each code, two bars are displayed side by side, the rightmost one (resp., leftmost one) representing the true percentage (resp. predicted percentage) of documents that have the code. The figure gives a compelling display of the quantification accuracy of our system.

These very good quantification accuracy results show that out system can reliably be used for quantification in operational situations. One such situation springs to mind, namely, the large-scale batch classification of legacy radiology reports for establishing temporal trends in the evolution of pathologies. The availability of an automatic system for doing this at a very good accuracy level makes this a plausible scenario.

6. Related work

In this section we review related work on the automatic classification of medical reports, showing the differences between these works and ours.

de Bruijn, Hasman, and Arends (1997) address the problem of the automatic classification of clinical text using the SNOMED classification scheme. In order to classify a new document the authors use the 1-NN nearest neighbour algorithm. The problem with this "lazy" approach is that 1-NN does not have an offline learning stage, and all the work is performed at classification time. This means that the algorithm is very inefficient, since for each test document it needs to find the maximally similar training document, which requires a number of similarity computations linear in the number of the training examples. Dreyer et al. (2005) try to automatically identify the presence of clinically important findings or the presence of recommendations for subsequent action in unstructured radiology reports. Although the authors use a combination of natural language processing and machine learning techniques, using decision trees as the learning device, they tackle a multi-label classification task, which is fairly different (and easier) than the "exactly-1" and "at-most-1" classification tasks that we need to address at each level of the two hierarchies. Wilcox and Hripcsak (1999) also address the problem of automatically classifying radiology reports. The authors tackle this problem using machine learning techniques to automatically determine the presence of six clinical conditions in chest radiography reports. Unlike in our work, the dataset used by the authors is really small (only 400 reports), and the classification algorithm is flat, while ours is hierarchical. Aronow, Fangfang, and Croft (1999) address the task of "ad hoc classification" of mammography reports. The problem addressed is completely different from ours, since in Aronow et al. (1999) the authors want to identify the most relevant reports *within a user query*; in our work we want instead to classify the radiology reports within a fixed hierarchical classification scheme.

Stanfill, Williams, Fenton, Jenders, and Hersh (2010) review a large body of work in the automated coding of clinical material, including radiology reports. They indicate that "(...) automating clinical coding is a difficult task, made even more difficult by the clinical texts that must be processed (...)", and conclude that "Further development of these systems and a better understanding of the tasks for which they will be used are needed before we can conclude that automated coding and classification systems meet performance standards adequate for use in complex clinical coding processes (...)". While no less than 113 papers are covered in their survey, none of them appears to use the ACR Index as the target

classification scheme. It has to be added that Stanfill et al. (2010) only addresses studies tackling clinical texts expressed in English. This is not very representative of the accuracy levels that can be obtained in languages other than English, since specialized lexical resources are far more abundant for English than for any other language, thus making the task of obtaining higher accuracy levels easier.

As a general comment, we should note that out work is probably the first study in the classification of medical reports to experiment at such a large scale, both in the number of medical reports used (more than 240,000) and in the size of the classification scheme used. We should also add that, from the standpoint of text classification, this paper probably presents the first experimental study in *single-label* (i.e., "exactly 1 code per document") text classification on a very large classification scheme. In fact, while it is true that experiments on very large classification schemes have been presented already (Bennett & Nguyen, 2009; Liu et al., 2005; Tang, Rajan, & Narayanan, 2009; Xue, Xing, Yang, & Yu, 2008), they concerned *multi-label* classification. In multi-label classification there is no reason, in principle, why effectiveness should deteriorate in moving from, say, a classification scheme consisting of 100 classes to one consisting of 100,000 ones, since each class is a binary classification task in itself, independent of the others. In other words, in multi-label classification the large size of the classification scheme is challenging from the point of view of *efficiency*, and not from the one of classification *accuracy*. In single-label classification, instead, the large size of the classification scheme is challenging also from the standpoint of accuracy, since picking *the* right class out of a very large pool of classes is of course more difficult than picking it from a small pool of candidate classes.

Finally we remark that, to the best of our knowledge, this is the first work that discusses the issue of quantification in the context of medical reports, and of its possible applications in the field of epidemiology.

7. Conclusions

In this paper we have reported our experiments on the automatic coding of radiology reports written in Italian under the ACR classification scheme. As discussed, correctly classifying data from the UmbertoI dataset is hard, since the UmbertoI reports are each associated to exactly 1 class, and picking *the* correct anatomic location class from a set of 719 legally assignable ones and *the* correct pathology class from a set of 5,269 legally assignable ones is like finding a needle in a haystack. Relative to the *a priori* difficulty of this task, our system has thus shown good classification accuracy. It was not possible to scientifically determine *how* good this classification accuracy is, due to the absence of data that would allow a thorough inter-coder agreement study to be performed. However, we think that the accuracy levels obtained ($F_1^{\mu} = 0.611$ for anatomic location and $F_1^{\mu} = .490$ for pathology) are not sufficient for envisaging fully automatic and unassisted coding, but are sufficient for making the use of the system cost-effective in scenarios involving semi-automatic, assisted coding, such as in recommending the correct codes to use for a given report, or in suggesting the correct codes for reports that the system deems likely to be incorrectly coded.

We have also reported the accuracy of our system at performing *quantification*, a recently defined task that merely requires the accurate estimation of the frequency of each individual code in the test set. Here our system displayed an excellent accuracy, with average error rates (measured in terms of the discrepancy between true and predicted frequency) of 0.05% or lower. These accurate estimations clearly allow a system such as this to be used in completely autonomous, unassisted coding for applications in the field of epidemiology, a discipline which is more concerned with estimating frequencies of occurrence than with assessing individual cases. A system such as ours, with coding speeds of more than 30 documents per second and quantification accuracy of 0.05% or lower, could reliably be used in automatically coding huge batches of legacy reports for the retrospective analysis of epidemiological trends.

8. Authors' contributions

AE and FS have jointly been responsible for designing the experiments. AE has been responsible for preparing the data, running the experiments, and preparing the tables with the results. FS has been responsible for writing the paper. AE and FS have jointly been responsible for critically revising the paper.

Acknowledgments

This work has been funded by NoemaLife SpA in the framework of the ConnectToLife project. Thanks to the Istituto di Radiologia, Politecnico Umberto I, Roma, IT, for making available the dataset used in this work; to Giulia Chiaruzzi and Cristiano Querzé for assistance in obtaining it; and to Gianpiero Camilli, Michele Carenini, and Davide Distefano, for encouraging this work.

References

American College of Radiology. (1992). Index for radiological diagnoses (4th ed.). Reston, US: American College of Radiology.

Aronow, D. B., Fangfang, F., & Croft, W. B. (1999). Ad hoc classification of radiology reports. *Journal of the American Medical Informatics Association, 6*(5), 393–411.

Bella, A., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. (2010). Quantification via probability estimators. In *Proceedings of the 11th IEEE international conference on data mining (ICDM 2010)* (pp. 737–742). Sydney, AU.

Bennett, P. N. & Nguyen, N. (2009). Refined experts: Improving classification in large taxonomies. In *Proceedings of the 32nd ACM conference on research and development in information retrieval (SIGIR 2009)* (pp. 11–18). Boston, US.

Chakrabarti, S., Dom, B. E., Agrawal, R., & Raghavan, P. (1998). Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *Journal of Very Large Data Bases, 7*(3), 163–178.

de Bruijn, B., Hasman, A., & Arends, J. W. (1997). Automatic SNOMED classification: A corpus-based method. *Computer Methods and Programs in Biomedicine, 54*(1/2), 115–122.

Dreyer, K. J., Kalra, M. K., Maher, M. M., Hurier, A. M., Asfaw, B. A., Schultz, T., et al. (2005). Application of recently developed computer algorithm for automatic classification of unstructured radiology reports: Validation study. *Radiology, 234*(2), 323–329.

Esuli, A., Fagni, T., & Sebastiani, F. (2006). MP-boost: A multiple-pivot boosting algorithm and its application to text categorization. In *Proceedings of the 13th international symposium on string processing and information retrieval (SPIRE 2006)* (pp. 1–12). UK: Glasgow.

Esuli, A., Fagni, T., & Sebastiani, F. (2008). Boosting multi-label hierarchical text categorization. *Information Retrieval, 11*(4), 287–313.

Esuli, A., & Sebastiani, F. (2010). Sentiment quantification. *IEEE Intelligent Systems, 25*(4), 72–75.

Esuli, A., & Sebastiani, F. (2010). Machines that learn how to code open-ended survey data. *International Journal of Market Research, 52*(6), 775–800.

Forman, G. (2008). Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery, 17*(2), 164–206.

Hersh, W., Buckley, C., Leone, T., & Hickman, D. (1994). OHSUMED: An interactive retrieval evaluation and new large text collection for research. In *Proceedings of the 17th ACM international conference on research and development in information retrieval (SIGIR 1994)* (pp. 192–201). Dublin, IE.

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. In *Proceedings of the 14th international conference on machine learning (ICML 1997)* (pp. 170–178). US: Nashville.

Langlotz, C. P. (2006). RadLex: A new method for indexing online educational material. *RadioGraphics, 26*(6), 1595–1597.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research, 5*, 361–397.

Liu, T., Yang, Y., Wan, H., Zeng, H., Chen, Z., & Ma, W. (2005). Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explorations, 7*(1), 36–43.

Meir, R., & Rätsch, G. (2003). An introduction to boosting and leveraging. In S. Mendelson & A. Smola (Eds.), *Advanced lectures on machine learning* (pp. 118–183). Heidelberg, DE: Springer Verlag.

Schapire, R. E., & Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning, 39*(2/3), 135–168.

Stanfill, M. H., Williams, M., Fenton, S. H., Jenders, R. A., & Hersh, W. R. (2010). A systematic literature review of automated clinical coding and classification systems. *Journal of the American Medical Informatics Association, 17*(6), 646–651.

Tang, L., Rajan, S., & Narayanan, V. K. (2009). Large scale multi-label classification via MetaLabeler. In *Proceedings of the 18th international conference on world wide web (WWW 2009)* (pp. 211–220). Madrid, ES.

Wilcox, A. & Hripcsak, G. (1999). Classification algorithms applied to narrative reports. In *Proceedings of the AMIA annual symposium* (pp. 455–459). Washington, US.

Xue, J. C., & Weiss, G. M. (2009). Quantification and semi-supervised classification methods for handling changes in class distribution. In *Proceedings of the 15th ACM international conference on knowledge discovery and data mining (SIGKDD 2009)* (pp. 897–906). Paris, FR.

Xue, G. -R., Xing, D., Yang, Q., & Yu, Y. (2008). Deep classification in large-scale text hierarchies. In *Proceedings of the 31st ACM international conference on research and development in information retrieval (SIGIR 2008)* (pp. 619–626). Singapore, SN.