Automated annotation of imaging experiments via multi-label classification

Matthew D. Turner [1,2,3,*], **Chayan Chakrabarti** [1], **Thomas B. Jones** [1], **Jiawei F. Xu** [1], **George F. Luger** [1], **Angela R. Laird** [4], **and Jessica A. Turner** [2,3,5]

[1] *Department of Computer Science, University of New Mexico, Albuquerque, New Mexico, USA*
[2] *Mind Research Network, Albuquerque, New Mexico, USA*
[3] *Conjectural Systems, Albuquerque, New Mexico, USA*
[4] *Department of Physics, Florida International University, Miami, Florida, USA*
[5] *Departments of Psychiatry and Psychology, University of New Mexico, Albuquerque, New Mexico, USA*

Correspondence*:
Matthew D. Turner
Department of Computer Science, University of New Mexico, Albuquerque, New Mexico, USA,
matthew.turner.phd@gmail.com

Research Topic

ABSTRACT

We provide baseline performance tests of common problem transformations and machine learning (text mining) methods applied to the problem of automatically classifying or labeling multi-label biomedical research literature. Label terms are from the CogPO ontology, the text corpora are abstracts of published papers, and the methods use the performance of a human expert's knowledge as training data. We aim to replicate the expert's annotation of labels for the experimental stimuli, cognitive paradigm class, response types, and other relevant dimensions of the experiments. Specific problem transformations tested are the binary relevance and label powerset methods. We use several standard machine learning methods: naive Bayes, k-nearest neighbors, and support vector machines (specifically SMO or sequential minimal optimization). Exact match performance ranged from only 15% in the worst cases to 78% in the best cases. This collection of results demonstrates what can be achieved with off-the-shelf software components, little to no pre-processing of raw text, and no serious attempts at optimization. Any more sophisticated methods for automated annotation of neuroimaging experimental methods would need to perform at or above the performance levels demonstrated here to be worth serious consideration as a practical method.

Keywords: text mining, data mining, multi-label classification, bioinformatics, CogPO, neuroimaging, annotations

1 INTRODUCTION

Scientific publication in cognitive neuroscience today is proceeding at an intense pace; a `pubmed.gov` search revealed that for the four year period 2009-2012, there were 5033 total publications tagged "human brain mapping," with the number of publications between 2009 and 2012 increasing by 12% each year. The situation is similar in other fields. We are faced with a deluge of new results and publications across all fields every year (Howe et al., 2008). This has created problems for data warehousing, searching, and curation. This latter term refers to the acquisition, selection, annotation, and maintenance of digital information.

The curation of this massive collection of scientific literature is a challenging problem. Although some tools exist to assist researchers with the management of this vast collection of data, most curation of scientific research literature is done in-house by the researchers themselves. Among the primary tools of curation are computer ontologies and controlled vocabularies (Trieschnigg et al., 2009). Controlled vocabularies limit language to terms with precise unitary meanings and ontologies replicate some of the logical structure of scientific language in a computable fashion, allowing researchers to more effectively search and process the scientific literature.

The BrainMap database (`www.brainmap.org`) was developed to provide a repository of the results from the human neuroimaging literature (Fox et al., 2005; Fox and Lancaster, 2002; Laird et al., 2005; Lancaster et al., 2005; Turner and Laird, 2012). The BrainMap schema developed as a way to describe PET and fMRI experiments and the conditions which led to the activation loci reported in the publications. This schema describes the subject groups included in the analyses (e.g. healthy controls and adults with autism), the context of the experiment (e.g. a pre/post treatment study), the behavioral domain being studied by each analysis (e.g. attention and memory), the specific paradigm class (e.g. memory for faces), and a set of terms and relationships for the experimental stimuli used in the conditions being contrasted in each analysis. The terms used to describe the the experimental conditions, their definitions and relationships, have been formalized in the Cognitive Paradigm Ontology (CogPO; Turner and Laird, 2012). The primary descriptors are a set of terms for the Stimulus Type (e.g. flashing checkerboard, tone, word, or picture), the Stimulus Modality (e.g. visual, auditory, interoceptive), the Instructions given to the subject (e.g. attend, discriminate, imagine), and the Response Type (e.g. button press, speech) and Response Modality (the part of the body used to make the response, e.g. hand, foot, face). Each experimental condition is a combination of these characteristics, and the loci of activation are commonly the result of comparing fMRI BOLD signal during one combination versus another (for instance, changing the stimulus type or changing the instructions while maintaining the same stimuli). The BrainMap project includes the database of papers and experiments as well as related software to both to find papers based on these terms (Sleuth) and to perform meta-analysis over the results from comparable experiments (GingerALE). This toolset has led the way in meta-analyses of fMRI and PET studies, identifying commonalities of brain activation across the literature on working memory, depression, and many other topics (Bzdok et al., 2012; Farrell et al., 2005; Fitzgerald et al., 2006; Laird et al., 2009, 2005; Menzies et al., 2008). The current database includes manually annotated results from approximately 2,298 publications—covering 10,924 experiments—and spanning the last 20 years of human neuroimaging research.

While these manual curation methods are useful, there is a bottleneck; given the rate of publication it is challenging the for curators to manually annotate the literature as it is produced. Coupled with this is the fact that there are very few people in the scientific community whose primary task is curation, and they are often lacking in the specialized knowledge required for making classifications using the specialized terms. Lastly, the scientists producing the literature themselves are often neither qualified to annotate their own work nor are they interested in the annotation task *per se* (Lok, 2010). A technological solution appears to be required and will require the use of machine learning tools.

The problem of ontology annotation, the marking up of scientific articles with terms and semantic structure based on an ontology, is related to a machine learning problem known as "multi-label classification." This is the most general form of the document classification task. The simplest form, binary classification, is the most well-developed area of automatic classification. In this task, learning machines are trained to determine if an instance (article) should be classified as being in a given class or not. We may think of this as determining if the instance has a label or does not; for instance, an article's content might be classified as "human brain mapping" or it might not. We are concerned with one choice and two options, either in the class or not in the class. Multi-*class* classification involves a set of classes that are mutually exclusive (every instance is in *at most* one class) and exhaustive (every instance is in *at least* one class). Here we are again concerned with a single choice, but there are more than two options. For instance, a newspaper article might be selected to be placed in the "sports," "business," or "local" section of the newspaper; each article to be printed must go into at least one section, and will appear in at most one section.

In multi-label classification, each instance classified will have some labels applied to it; the set of labels is not necessarily mutually exclusive or collectively exhaustive, and *a priori* we do not know which or how many labels a given instance may receive. An example of this is a newspaper's website. While articles can appear only in one section of a printed paper, on the website an article may be tagged with several sections. So an article on the financial situation of a sports team may be labeled "sports" and "business" and a story about a local restaurant sponsoring a local high-school football team might very well be labeled "business," "local," "sports," and "food." Binary and multi-class classification can be considered as special cases of, or restrictions on, the multi-label problem. The multi-label problem has been growing in importance as the internet has made larger pools of content available with no single classification scheme. For a overview of multi-label classification, see Tsoumakas and Katakis (2007) and Tsoumakas et al. (2010); for an overview of the technical issues involved, see Madjarov et al. (2012).

Recently there has been an increase in the application of machine learning methods to biomedical literature analysis. Many of these approaches seek novel algorithms to solve these problems. However, the machine learning literature is replete with well-established methods for binary and multi-class problems that perform quite well. Additionally, there are a number of methods to transform multi-label problems into one of these more restrictive forms described above. Before developing entirely new algorithms, it is reasonable to ask whether or not the tools at hand can achieve useful results. Additionally, the application of these methods may indicate where the issues in multi-label biomedical classification lie.

We seek to establish a baseline point of comparison for methods that may be developed for automated annotation of research abstracts using neuroimaging experimental terms. Here we apply entirely off-the-shelf solutions to the task of classifying scientific abstracts using the CogPO ontology. We present the methods in more detail than is perhaps common in the text-mining community, in service of making these results more repeatable by others, and to present these methods to neuroimaging researchers interested in automated annotation who may not otherwise be aware of them. The performance characteristics here may be viewed as a reasonable minimum performance point, which must be exceeded by new or more complex algorithms if they are to be viable competitors for practical applications in this arena.

2 MATERIAL & METHODS

2.1 DATA

We have 3 corpora and 7 label dimensions to evaluate on each text corpus. Each corpus consists of 247 biomedical abstracts retrieved from MEDLINE, based on a selection of the papers included in BrainMap, and which covered all the studies from a selection of disorders for which we possessed expert annotations. The label dimensions were not otherwise constrained; these are discussed in section 2.1.2.

2.1.1 Corpora The 247 MEDLINE abstracts are basis for the training and testing instances for the machine learning algorithms. The features or attributes to be used for classification were vectors indicating the presence or absence of certain words in the abstract text. The plain abstract text forms the first corpus, and two additional corpora were created by elaborating this basic text in two ways. First, with the addition of titles and medical subject heading (MeSH; `www.nlm.nih.gov/mesh/`) keywords as additional features; then these were passed through the NCBO annotator (`bioportal.bioontology.org/annotator`) to add annotations from several ontologies (not including CogPO) to determine if these markups would improve CogPO classification performance. This produced the 3 corpora:

1. **Plain.** The original text of each paper's abstract.
2. **Title and Keywords.** The text of each paper's abstract with MeSH keywords and article titles added.
3. **Ontology Annotated.** The "title and keyword" abstracts with the additional annotation of ontology terms.

Note that this is a progressive hierarchy, with each level containing the features that came before it.

The MeSH labels were limited to the "descriptor names" without the "qualifier names". The ontologies used for annotation were the Foundation Model of Anatomy (Rosse and Mejino, 2003), Cognitive Atlas (Poldrack et al., 2011), NIFSTD (Bug et al., 2008), and RadLex (Langlotz, 2006). The goal was to annotate the brain areas, other cognitive terms, or imaging methods that might have been mentioned in the abstract text. The NCBO Annotator leverages the structure of the NCBO ontologies to annotate text with generalizations of matching terms; i.e., if a word in the text being annotated matches a term in an ontology, the Annotator can also return the superclass(es) of the matching terms, to provide more general concepts. The ontologies used here were often very flat, though, without many levels available in the hierarchy, and thus only terms from the level matching the abstract text was included. For example, taking more than the matching level led to annotation with very abstract philosophical terms such as "dependent continuant" from the foundational ontology BFO (`www.ifomis.org/bfo`) that forms the base for several of these ontologies. In essence, there was a short path from specific terms to excessively abstract terms.

The abstract text was directly tokenized based on whitespace and punctuation, making each individual word into a token. This process also made numbers into tokens; the numbers were sometimes broken into multuple tokens (e.g. 0.5 became 0 and 5). No attempt was made to apply semantic mapping or concept identification to the original abstract text; each abstract word was treated as a single feature even when it should have been part of a multi-word token. Many of the MeSH labels and ontology annotations were also multi-word constructs, such as "Tomography, Emission-Computed." In this case, we preserved the underlying concept by mapping these to single tokens. We were able to do this because the MeSH and ontology queries returned the multi-word concepts with delimiters, allowing their preservation.

Each abstract was reduced by stopword removal, using the Natural Language Toolkit (NLTK; `nltk.org`) English stop word list (Bird et al., 2009; Bird and Loper, 2004). These were then converted to a "bag of words" vector representation with WEKA (Hall et al., 2009). Only the presence or absence, 1 or 0 respectively, of each word was recorded. In some applications, the term "bag of words" is reserved for vectors of counts; in this work the vectors are binary presence/absence representations. It should be noted that only basic English stop words were removed. No effort was made to remove numbers (meaningless in a "bag of words context"), specialized biomedical terminology occurring either too often or not often enough to be discriminating, and any other low-information vocabulary

This produced for each corpus a collection of 247 instance vectors, one for each abstract, each of a length equal to the length of the dictionary for the corpus. Each of the three corpora had a different dictionary length. The plain text corpus had a dictionary length of 3603 words, while the keyword/title and annotated abstracts had dictionaries of 3918 and 3919 words respectively. There is a substantial overlap between the ontology annotator's results and the previously applied MeSH headings and base vocabulary of the abstracts. It should also be noted that the bulk of the features come from the author written abstract text, not the keywords or annotations (approximately 92%).

2.1.2 Labels The labels for each abstract came from the expert assignment of CogPO terms to the corresponding scientific papers. The expert used the full-text of the papers to make the assignment; thus they had access to more information than was contained in the input to the machine learning algorithms. CogPO provides a number of dimensions of labels, as described above in the BrainMap schema. We used the following dimensions: *behavioral domain*, *cognitive paradigm class*, *instruction type*, *response modality*, *response type*, *stimulus modality*, and *stimulus type*. The number of labels present in each dimension range from 5 to 48; see **Table 1**. The number of labels per dimension reported here are the numbers actually present in this particular sample of abstracts; CogPO has additional labels not used here in our available instances. Given our methods, labels without any instances would automatically drop out, so we can restrict the analysis to just the labels present without any loss of generality.

Additional label characteristics presented in **Table 1** are as follows. A standard measure in multi-label classification is label cardinality, the number of labels per instance. For multi-label data sets this varies by instance, and is usually reported as an average summary measure; here we present this usual average label cardinality as $\mathrm{LC}_{\mathrm{avg}}$. We also include the maximum number of labels applied to a single instance, $\mathrm{LC}_{\mathrm{max}}$; e.g. in the case of Behavioral Domain, at least one abstract was annotated with 8 different terms, but the average number of labels was 1.846. The minimum ($\mathrm{LC}_{\mathrm{min}}$) is always 1. The measure $\mathrm{P}_{\mathrm{UNIQ}}$ for multi-label corpora is defined in Read et al. (2011), and is the number of unique label sets divided by the number of instances. Finally, $\mathrm{P}_{\mathrm{min}}$, is the proportion of the data that is assigned the minimum number of labels, which for all of our dimensions is 1 label,

$$\mathrm{P}_{\mathrm{min}} = \frac{|\{\text{Instances with 1 label}\}|}{N}$$

i.e. the number of instances with one label divided by the total number of instances. We use this measure instead of the $\mathrm{P}_{\mathrm{max}}$ measure also defined in Read et al. (2011); in our case we felt this was more revealing. For our data $\mathrm{P}_{\mathrm{max}}$ is always based on 2 cases ($\mathrm{P}_{\mathrm{max}} = 0.0081$; for both stimulus modality and response modality dimensions) or 1 case (0.0041; all other dimensions). Note that $\mathrm{P}_{\mathrm{min}}$ shows that the modal number of labels for each dimension is 1; the median number of labels is 1 as well, for all dimensions, except for behavioral domain where it is 2.

Table 1. Characteristics of the data by dimension of the CogPO ontology and label sets. See text for details. The last column is the value of k for each label set when given to the kNN algorithm, see section 2.3.2.

Dimension	# Labels	$\mathrm{LC}_{\mathrm{avg}}$	$\mathrm{LC}_{\mathrm{max}}$	$\mathrm{P}_{\mathrm{UNIQ}}$	$\mathrm{P}_{\mathrm{min}}$	*k*
Behavioral Domain	40	1.846	8	0.429	0.413	9
Cognitive Paradigm Class	48	1.291	4	0.336	0.761	9
Instruction Type	14	1.648	6	0.251	0.510	25
Response Modality	5	1.308	3	0.036	0.700	21
Response Type	9	1.324	4	0.069	0.696	11
Stimulus Modality	5	1.150	3	0.036	0.858	25
Stimulus Type	17	1.494	4	0.247	0.587	9

2.2 PROBLEM TRANSFORMATIONS

A problem transformation is any method that transforms multi-label data into a collection of single-label (binary) classification problems or which reduces a multi-label problem to a multi-class problem (Cherman et al., 2011; Modi and Panchal, 2012; Read et al., 2009, 2011; Santos et al., 2011; Tsoumakas et al., 2010). Here we consider two problem transformation methods: binary relevance (BR) and label powerset (LP; also referred to as LC for "label concatenation"). These methods are often implicitly incorporated into other methods. The benefit of abstracting out the transformations is that it allows new applications to be constructed easily by recycling binary and multi-class methods. In any use of a problem transformation method, both the transformation and the underlying classifier it is combined with must be indicated to have a complete specification.

Notation Assume for the following that L is a set of labels for a given problem, $|L|$ represents the size of the set L (i.e. number of labels), and λ stands in for an individual label as required. So, $L = \left\{\lambda_1, \lambda_2, ..., \lambda_{|L|}\right\}$. We let $\bar{\lambda}$ stand for the complement (negation) of λ. Following the literature, the set of instances will be called D and we will let N represent the number of instances in the training set, so: $N = |D|$. We let d represent the number of features of the feature space. Here d will equal the number of words in the dictionary and will vary by corpora.

2.2.1 *Binary relevance*

The binary relevance (BR) method reduces a multi-label problem to collection of binary classification problems. It does this in the simplest and most obvious way; BR gives each label has its own classifier. For a problem with $|L|$ labels, a separate classifier is built for each λ and, for a given classifier, each abstract is identified as either having the label λ or not, labeled $\bar{\lambda}$. This reduces the $|L|$-label problem to $|L|$ statistically independent binary problems, each with appropriately re-coded data. Therefore, any binary classifier may be applied to multi-label data.

For binary classifiers that produce probability or confidence estimates for each label, a threshold, t can be chosen for inclusion of that label in the multi-label classification of that instance. A threshold calibration procedure can be used to automatically select this value; a numerical grid search is conducted for values of t which match the predicted data set's average label cardinality to that of the training set (Fan and Lin, 2007; Read et al., 2011). This procedure is simple and efficient and empirically better justified than the arbitrary selection of a value for t. It is used when BR is combined with naive Bayes and k-nearest neighbors below.

The problem with this method is clear: dependencies among the labels are ignored, as each is classified separately. However, the method is simple, both computationally and conceptually, and scales linearly with the number of labels $|L|$ (Read et al., 2011); overall computational complexity will depend on the underlying classification algorithm. It is resistant to over-fitting, it does not require examples of every possible label combination and the models built for each label are independent of one another which allows updating of labels without having to completely recompute all the other models (Read et al., 2011). This is important for on-line or continually updating systems. Additionally, the assumption of independence among labels is similar to that made by naive Bayes regarding features (see below),and that method often works well-despite the assumption (Zhang, 2004, 2005). One would expect that more customized methods that can improve performance will make use of these dependencies.

2.2.2 *Label powerset*

The label powerset (LP) method reduces a multi-label problem to a single multi-class problem. Under LP, each abstract's unique label combination is reduced to a single, corporate, label. With this method there will be as many labels as there are unique combinations. So an instance that is classified as λ_1 and λ_2 would receive the single combined label λ_{12}. (We can assume a default label ordering on L such that λ_{12} and λ_{21} will be the same.) Thus the collection of labels for each instance is reduced to a single label that is the concatenation of all the labels assigned to the instance.

For instance, for the behavioral domain label dimension we have 40 labels, appearing in 106 unique combinations (in **Table 1**, this number is $\mathrm{P}_{\mathrm{UNIQ}} \times 247$ or $\mathrm{P}_{\mathrm{UNIQ}} \times N$). From the point of view of the underlying classifier, this is a single classification with 106 mutually exclusive and exhaustive classes; each abstract is assigned to exactly one of the classes. Thus, any classifier that can be applied to a multi-class classification can be used. It is worth emphasizing that most binary classifiers have extensions to the multi-class problem already, so this transformation still allows a full range of off-the-shelf components to be used.

Under LP, a single classifier is built, and if this classifier assigns probabilities or confidences for each abstract to be assigned to each of the 106 unique combinations, then the largest of these confidences is selected as the label combination. The underlying classifier simply reports the class selection, then that is used; there is no thresholding process as for BR.

Potential problems with this method are complexity and over-fitting. The computational complexity of this problem is a function of how the underlying learning algorithm handles the number of classes in a multi-class problem, but the worst case scales exponentially with $|L|$, although this is constrained by the amount of data, $\min(N, 2^{|L|} - 1)$, see Read et al. (2011). However, for realistic cases this may be within a usable tolerance; our behavioral domain label set with 40 labels has a worst-case complexity of 10^{12}, but both the number of actual label combinations (106) and size of the data set (247) severely restrict the problem to realistic computational requirements, here 10^2 in either case. However, this matter is an empirical question and there may be data to which LP cannot reasonably be applied.

This method is very sensitive to the specific label combinations in the training data; it only learns the label combinations that are present, a kind of over-fitting. Thus, if new data are analyzed, with new label combinations not present in the training data, either the entire model will have to be retrained with new data, or the model without those combinations in the training data will never be able to specifically predict the new combinations.

2.3 MACHINE LEARNING ALGORITHMS

Once a problem transformation has been applied to the data, a machine learning algorithm must be used on the transformed data. Here we consider 3 standard methods: naive Bayes, k-nearest neighbors, and a type of support vector machine called sequential minimal optimization. These methods are relatively simple, easily available off the shelf, and are known to work well in a variety of machine learning and text mining contexts. For notation, see the beginning of section 2.2.

2.3.1 Naive Bayes Naive Bayes (NB) is a standard machine learning algorithm that is often used as a first approach for new problems (Eyheramendy et al., 2003; McCallum et al., 1998; Rennie et al., 2003; Witten et al., 2011; Zhang, 2004); NB is often quite effective. The method uses Bayes' theorem to transform the label-conditional probabilities, $P(\text{feature}|\lambda)$, derived from the training set, into $P(\lambda|\text{feature})$, the conditional probabilities of an instance having label λ given the presence of a feature. These probabilities, for each feature present in an instance, are combined to produce an estimate of the probability of the instance having label λ. The "naive" in the name refers to the assumption of feature independence present in the model. To make the calculations tractable, features are treated as statistically independent, usually an unreasonable assumption for real data. Mathematically this means that the probability of an instance having a label, $P(\lambda)$ is the product of the $P(\lambda|\text{feature})$ values for features present in the abstract, and the compliments of these for features not present in a given abstract. Despite the logically unreasonable independence assumption, this method works quite well in most applications (McCallum et al., 1998; Zhang, 2004, 2005), but see Rennie et al. (2003).

For binary classification, as under the BR method, the NB classifier for each label will return a probability for that label only. A threshold probability, t, can be chosen iteratively as described in section 2.2.1 on the BR transformation. For the LP method, a single NB classifier is built that returns a probability distribution across the unique label combinations. In this case, the label combination with the highest probability is chosen as the label combination for a new instance.

The NB classifier has no "tunable" hyperparameters affecting its performance. In that regard it is usually viewed as being data-driven.

2.3.2 k-Nearest Neighbors We implemented the k-nearest neighbors (kNN) classifier under both BR and LP; see Spyromitros et al. (2008) for a discussion of these methods. In kNN, the k nearest neighbors to the instance to be classified are found in feature space. For this to be meaningful, a definition of distance over the feature space must be adopted. We chose to use the Euclidian distance, as that tends to be a common default and is available off-the-shelf. Note that the distance between instances is computed in a very high dimensional space; each corpus' dictionary defines the dimension of the feature space. For example, in the plain corpus with 3603 tokens, the distances are computed between points in a 3603-dimensional feature space. This distance will be equal to the square root of the number of mismatched words in the two abstracts being compared; more mismatches means greater distance. Words not present in either abstract or words present in both do not affect the distance.

Once the k neighbors are found, their label frequencies are analyzed. Under the BR transformation, a confidence for each label is generated and a cutoff threshold, t is chosen as above. For the LP transformation, the most common unique label combination of the k neighbors is selected. For an alternate method that uses kNN internally, see Zhang and Zhou (2007).

The performance of kNN can be degraded by a variety of issues: noisy features, the presence of irrelevant features, or scaling of the feature values. The last of these is not a problem in our presence-absence approach (see section 2.1.1) as each feature is represented on the same scale. However, the large number of features surely presents many irrelevant features for classification, and there are terms used in vague, overlapping, and ambiguous ways in the abstract texts, so both of the other issues are present in this type of data.

For kNN, there is one hyperparameter, k, the number of neighbors to consider. Despite the importance of selecting a good value for k, or for selecting hyperparameters more generally, there is not a large body of research literature on this topic. For k, we chose to execute a comprehensive grid search for all values of k from 1 to N on the plain abstract corpus. We optimized for log-loss, a criterion which penalizes errors based on confidence and therefore rewarding conservative prediction (Read et al., 2011). The k determined in this fashion is consistent with the optimization of other evaluation measures, such as F_1-micro (see section 2.4). See the last column of **Table 1** for the best value of k

for each label dimension. For the kNN analyses in our results, we used this optimum k for each dimension; it is generally believed that this form of hyperparameter selection is overly optimistic, so the kNN results should be interpreted with this in mind.

2.3.3 Sequential Minimal Optimization Sequential minimal optimization (SMO) is one of a class of learning algorithms called support vector machines (Platt, 1998). These algorithms have been shown to perform well in text mining applications (Cohen and Hersh, 2005). Support vector machines are a type of hyperplane classifier that seek out hyperplanes that distinguish classes (labels) in the feature space. This is done in such a way that the margin or distance between the boundaries of the classes in the feature space are maximized (so-called maximum margin classification). The methods are called "support vector" machines because a set of vectors lying on the boundaries (the support vectors) are found. Other feature vectors can be changed arbitrarily without changing the classification performance. These methods can be used with non-linear transformations (kernels) but for our feature space with dimension always greater than 3600 (see section 2.1.1) we can use the linear kernel. The assumption with these methods is that in such a high dimensional space you can find the required hyperplane even without a non-linear transformation.

When using the linear kernel with SMO, there is only one hyperparameter to set, the *complexity*, C. This parameter restricts the search space for solutions to the optimization problem; for details, see Platt (1998). We optimized this parameter via a numerical grid search. After extensive work on this, we discovered that the default setting for the WEKA software ($C = 1$) works very well for all dimensions and across all corpora, so this C was used for all experiments.

2.4 EVALUATION METRICS

The assessment of algorithm performance in the multi-label problem is substantially more challenging than in the single-label case (Madjarov et al., 2012; Tsoumakas et al., 2010). When an algorithm assigns a set of labels it may assign too few, missing some correct labels that should have been assigned or it may assign too many, adding some irrelevant labels. For any given label, we may easily determine the status, correct or incorrect; but for the entire set of assigned labels the usual case is some labels will be correct, some may be wrong (should not have been assigned), and some that should have been assigned are missed entirely. Evaluating bulk performance, over many labels and many instances is challenging for those reasons as well as the issues related to how the evaluation metrics are to be averaged. Unfortunately there is no single best measure of performance or universally agreed upon set of metrics.

In evaluating our results we used two measures: exact match (also called subset accuracy) and F_1-micro. Exact match is a very conservative measure of performance; it is simply the percentage of instances which are completely correctly labeled. Any missing, incorrect, or extra labels result in an instance being labeled as incorrect. The measure runs from 0 to 100 percent.

F_1-micro can be formulated as a measure of accuracy that is an average of precision and recall:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

where this is the scaled harmonic mean of the two. Precision measures if the labels returned are relevant to the instance, while recall measures the proportion of relevant labels that the algorithm returns out of the total correct labels for an instance. For more details, see Tsoumakas et al. (2010). Missing labels, extra labels, or incorrect labels all reduce the F-score, while correctly chosen labels increase the score. Note that this is the micro-averaged and instance based version of the F measure. This is commonly used when comparisons across data sets are relevant. The best possible F_1 score is 1 and the worst is 0, but it is not simply a proportion correct, as that concept is not uniquely defined in the multi-label scenario.

2.5 SOFTWARE AND SOURCES

All of the experiments conducted in this paper were completed using the MEKA software package (`meka.sourceforge.net`), the multi-label extension of WEKA (`www.cs.waikato.ac.nz/ml/weka/`). MEKA implements the problem transformation methods and allows the use of WEKA classifiers for the machine learning methods. We used MEKA's BR and LP (called LC in WEKA) problem transformations and WEKA's implementation of naive Bayes, kNN (called IBk), and SMO methods. For the problem transformation methods, naive Bayes, and SMO we used the default settings; for IBk we used the values of k reported above for each data set. Additionally, we used the default Euclidian distance function for kNN and the linear kernel for SMO.

The expert assigned labels for these abstracts have graciously been made available by the BrainMap staff. The authors can be contacted to obtain them for future research. The actual text of the corpora are from PubMed and, as such, are subject to copyright constraints that vary by journal; therefore our data sets cannot be made freely available by the authors of this paper. However, all of the abstracts can be readily downloaded from Pubmed by running a simple Eutils query. The authors will provide a list of MEDLINE abstract numbers or scripts to execute the query to interested parties.

3 RESULTS

3.1 EXPERIMENT 1: TRANSFORMATION AND ALGORITHM COMPARISON

In the first experiment, we directly compare the various combinations of problem transformation method and machine learning algorithm on the plain abstract text corpus for each of the 7 CogPO label dimensions. The basic results are presented in **Table 2**; organized first by

Table 2. Performance of SMO, NB, and kNN under the two problem transformations, label powerset (LP) and binary relevance (BR). Decimals are F_1-micro scores and percentages are exact matches. See text for details.

Dimension	Label Powerset			Binary Relevance		
	SMO	NB	kNN	SMO	NB	kNN
Behavioral Domain	0.413 29.4%	0.374 25.0%	0.285 14.6%	0.437 24.1%	0.537 23.3%	0.350 08.5%
Cognitive Paradigm Class	0.460 43.2%	0.404 37.5%	0.198 18.2%	0.416 28.3%	0.464 34.7%	0.281 12.1%
Instruction Type	0.485 36.1%	0.475 36.5%	0.379 26.4%	0.494 25.9%	0.538 23.9%	0.508 22.3%
Response Modality	0.741 54.2%	0.733 51.0%	0.636 48.2%	0.740 47.4%	0.744 49.8%	0.698 41.7%
Response Type	0.704 51.4%	0.689 51.8%	0.617 42.0%	0.702 44.5%	0.715 46.5%	0.641 31.5%
Stimulus Modality	0.838 78.1%	0.842 78.1%	0.741 68.1%	0.816 74.9%	0.814 72.4%	0.768 65.2%
Stimulus Type	0.439 30.7%	0.444 32.7%	0.303 17.8%	0.387 21.0%	0.478 20.6%	0.364 14.5%

transformation and then by learning algorithm within transformation. The rows in the table are the label dimension and the columns represent the results for the three methods SMO, NB, and kNN. The three columns on the left are LP transformed and the three on the right are BR transformed. In each cell, the upper number is F_1-micro (as a decimal) and the lower number is the exact match percentage. All the values reported in the tables are average estimates obtained from 10-fold cross-validation.

Reviewing the table shows some patterns. First, the exact match scores are uniformly greater for the LP transformation over the BR. This is not surprising as the LP transformation treats each unique combination of labels as a distinct entity, so it should be better at exact matches. However, as mentioned above, this leads to a type of overfitting: LP based multi-label classifiers cannot predict novel combinations of labels. Therefore, this increase in performance comes at a price; situations where novel combinations arise frequently will be a problem for this method.

Second, among the machine learning methods, there is no unambiguous winner. However, there is a loser. The kNN method, at least under our settings, is never the best method for these data. Under LP, kNN is always the worst method, under both the F_1-micro and exact match criteria. For the BR transformation, it does almost as badly; it only manages to be the second ranked method once (F_1-micro; Instruction Labels). It is worth noting that the kNN results are not always so terrible as to be unusable, but the method does sometimes fail dramatically when compared to the other methods.

Comparing performance across transformation methods, each learning algorithm against itself, we see that binary relevance is the clear winner. Both kNN and NB do better under BR than under LC with kNN always doing better and NB doing better in 6 out of 7 dimensions. SMO does better with LP in 5 out of 7 cases, however in two of those cases the difference in F_1-micro is ≤ 0.002. Given the fragile nature of LP compared to BR, this makes a good case for BR as the preferred basic problem transformation method.

Finally, turning to overall best performance, under F_1-micro the clear algorithm winner is NB (all cases) and BR 6 out of 7 cases (only the stimulus modality labels were better classified using LP). For exact match as a metric, as already mentioned LP is the better transformation. However, SMO and NB both performed well for some cases and less well for others; NB was the better method for 3 dimensions, SMO for 3, and one dimension (stimulus modality) was a strict tie. See the discussion for more on this.

3.2 EXPERIMENT 2: CORPORA COMPARISONS

The second question addressed here is whether or not the enhancement of the corpora with more features, selected from the MeSH headings and the ontologies as described above (section 2.1.1) improves classification performance. We extensively explored the combinations of methods and transformations, but only one representative case, NB under BR, will be presented here, in **Table 3**.

As should be clear from the table, there is no obvious improvement in performance obtained by adding the MeSH or ontology terms to the base text of the abstracts. Perhaps more surprising, the addition of titles in the title-keyword enhanced abstract texts does not show any obvious improvement. This is likely due to the abstract text already containing the critical elements of the title. Note also that the addition of ontology terms does not change the feature space (as noted in section 2.1.1).

Not presented here are the variability estimates from the 10-fold cross-validation procedure. These were uniformly higher in the title-keyword condition, with increases ranging from 0.001 to 0.021 (worst case; behavioral domain labels) and an average increase in variability of 0.011 F_1-micro units. This increase of variability, without concomitant increase in F_1-micro performance, is likely due to the increase of irrelevant features in the feature space as MeSH and title terms are added.

Table 3. Cross-corpora comparison experiment. Table presents F_1-micro values (see text) for naive Bayes under the binary relevance transformation across the three corpora.

Dimension	Plain	Title (MeSH) Keyword	Ontology Annotated
Behavioral Domain	0.537	0.534	0.534
Cognitive Paradigm Class	0.464	0.471	0.471
Instruction Type	0.538	0.534	0.534
Response Modality	0.744	0.745	0.745
Response Type	0.715	0.706	0.706
Stimulus Modality	0.814	0.815	0.815
Stimulus Type	0.478	0.496	0.496

4 DISCUSSION

We present performance characteristics for reproducing expert annotations of a human neuroimaging corpus of manuscripts, using the abstracts of the papers alone and an array of commonly-available multi-label classification techniques. Using an exact match criterion—how often does the method return exactly the labels that the human expert applied to the paper, no more and no less—the label powerset method does the best, in the easiest condition performing above 78%. However, while exact match is easier to interpret, F_1-micro is a better measure for evaluating performance overall as it does not completely penalize partial matches as complete misses. Using this as a criterion, we conclude that the combination of binary relevance and naive Bayes is the best performing combination across the data sets overall.

There is no universal or standard scale for comparing and interpreting F_1-micro; there are only relative comparisons. However, to provide some context we examine the results of Trieschnigg et al. (2009). There, six classification systems were compared in terms of their ability to assign MeSH keywords to abstracts, a similar task to ours. In the F_1-micro scores reported there, one system, the MTI or Medical Text Indexer, obtained a score of 0.4415; the authors consider this as "...perform[ing] quite well on the classification task" (Trieschnigg et al., 2009, p. 1216). Note that the MTI is production software, in practical use, and was the standard they used to compare against other systems. Our hardest label dimension, cognitive paradigm class, in experiment 2 is at about this level of performance and our other label dimensions exceed this (**Table 3**). This suggests that our classifiers are performing reasonably well over all the dimensions. Other results from Trieschnigg et al. (2009) are comparable to our first experiment, although we note that in their study k-nearest neighbors (kNN) was viewed as more successful algorithm. This is not surprising as the full specification of the variant of kNN they used is not given and our kNN algorithm achieves comparable performance in several label dimension and problem transformation combinations (**Table 2**). Finally, we mention that the task of finding MeSH terms is not completely comparable to our task, their formulation was not as a strict multi-label classification task, but their work suggests that F_1-micro scores in the 0.4 or greater range provide practically useful results.

Turning now to the features of the data that affect performance, we see that the performance varied tremendously based across the different label dimensions (compare the rows of **Table 3**) while performance was not significantly affected by changing the feature space (i.e. the corpora; compare the columns).

Performance across all methods was best for stimulus modality and response modality, which had the fewest labels (5 each), and were among the highest P_{min}, or proportion of instances with only a single label. The stimulus modality dimension was the highest P_{min}, while both stimulus and response modality dimensions were among the top 4 P_{min} values. The performance for response type was also notably higher than in the other dimensions, with fewer than 10 labels to choose from and 70% of the instances having only a single label. This suggests that a simpler label structure improves performance.

Performance dropped off dramatically with either increasing LC_{avg}, the average number of labels per instance, or with increasing $|L|$, the number of labels in L; the worst performance (**Table 3**, F_1-micro, plain abstracts column) was for cognitive paradigm class, stimulus type, instruction type, and behavioral domain (in order of increasing performance). These were the dimensions with the largest label sets. Both stimulus type and behavioral domain also had a larger proportion of instances with multiple labels ($1 - P_{\text{min}}$), but cognitive paradigm class had a surprisingly large proportion of single-label instances, and yet performed poorly. Comparing these with Trieschnigg et al. (2009), we still see practically acceptable F_1-micro performance.

It is worth noting that in the MeSH markup task in Trieschnigg et al., the test set was 1,000 abstracts with a label set of 3,951 MeSH terms; two orders of magnitude larger than our largest label dimension. Comparing those results with ours suggests that F_1-micro may be a function of the number of labels $|L|$ or possibly some scaled version of this. Unfortunately, neither Trieschnigg et al. (2009) nor Trieschnigg (2010) provides an exact number for the size of the training sets used for their kNN classifier, so we cannot make that comparison. However, they appear to have used large sets, with "at most" 1000 citations per MeSH term (Trieschnigg et al., 2009). It is important to contrast this with the number of training/testing instances we used which was 247 total. This suggests that relatively high performance may be achieved with very limited data (instances) given the richness of the feature space derived from abstract text.

An issue with the binary relevance method is that it assumes statistical independence of the labels assigned. Informal discussions with the human expert annotators has indicated that this is not the case in practice. For instance, the cognitive paradigm class label "go/no-go" implies a task that has the stimulus modality label "visual," response modality "hand," and response type "button press." This implication is not logically necessary (it is possible that it be otherwise) but for the papers in the BrainMap database, this implication is, effectively, certain. Additionally, there are logically necessary dependencies; for example, a "flashing checkerboard" (stimulus type) is necessarily presented to the "visual" stimulus modality. Expert annotators use both of these types of dependency knowledge in their label assignment task. None of the methods tested here can use this information explicitly.

There are off-the-shelf methods that might address this dependency issue. One method we tested was the classifier chains (CC) method Read et al. (2009, 2011), but we obtained no performance improvement with CC methods for our corpora. This was due to the presence of more than 3600 features (words or tokens) in the smallest of our corpora; many of which were irrelevant to classification. In ongoing work we are exploring methods for transforming the corpora to remove these irrelevant features. If this can be achieved, we would expect CC to efficiently improve overall performance of the classification process, but see Madjarov et al. (2012). Other transformation methods, not within the scope of this baseline analysis but certainly worth consideration, include pruned sets (Read et al., 2008), RAkEL (Tsoumakas et al., 2011); see (Santos et al., 2011) for a list. See Madjarov et al. (2012) for a substantial and recent review of this literature and comparison of the performance of many of these methods on other corpora. These are more much complex approaches, some of which include statistical and logical dependency information. We expect that they may lead to improved performance in combination with feature reduction methods.

A challenge for these techniques is the flexibility to handle new instances as they arise in new data; in the neuroimaging literature, new experimental paradigms arise frequently, and the CogPO terminology is expected to grow. This growth will be (1) in the addition of new terms for novel paradigms and (2) in the introduction of more precise terms as the granularity of the system moves from coarse to fine grained. BrainMap itself has already undergone several additions to the original term lists prior to the development of CogPO, with old terms being refined into several new terms. Each time new terms were included, it required a re-labeling of many experiments, to make sure their annotations are consistent with the updated label lists. This process will continue as research in these areas continues, cognitive experiments become ever more refined, new subdivisions of behavioral domains or cognitive processes come into vogue, and so on.

This is a problem for the label powerset transformation method; it is fragile with respect to label combinations. It cannot correctly label an instance which has a novel combination of annotations without retraining its underlying classifier on explicit examples of the new label combination. Thus, while this method had an advantage over binary relevance in the exact match measures, given the issues with extending the label powerset approach to the ever-expanding scientific literature—with the constant influx of new label combinations—its modest advantage over binary relevance is not sufficient to recommend it, at least not as a singular solution. However, binary relevance has the reverse problem, it cannot specifically model combinations of labels that carry the contingent or conditional information discussed above, and so its advantage in being less fragile is somewhat offset by this loss. While binary relevance is the better method given the present constraints, we anticipate future methods that combine the benefits and offset the losses of each of these methods when used as pure methods.

Besides the transformation approaches and classifier algorithms, the structure of the corpus and the structure of the label sets play a role in the ability to perform automated annotation. **Table 4** shows two of our data sets, compared with three other standard data sets used in multi-label classification. These data sets are ordered by complexity, which is usually defined as $N \times |L| \times d$; the product of the three relevant set sizes: instances, labels, and features. As shown, relative to other non-biomedical corpora commonly used for multi-label textmining research, our data sets fall toward the lower end of the complexity scale. We include the two extreme complexities for our various sets: "StimModPlain" is plain abstract with stimulus modality labels, the least complex of our sets; "CogParaAnno" is cognitive paradigm labels with annotated abstract text, the most complex. The other combinations lie between these extremes.

Table 4. Characteristics of several multi-label data sets compared with ours. Values taken from Read et al. (2011); see there for details and sources. For notation, see section 2.1.2 and 2.2. Included are the values for the least and most complex data sets included in this paper.

Name	Complexity	N	$\|L\|$	d	d/N	LC_{avg}	P_{UNIQ}	P_{max}
[1]StimModPlain	4449705	247	5	3603	14.59	1.15	0.036	0.008
[2]CogParaAnno	46463664	247	48	3919	15.87	1.13	0.336	0.004
Medical	63770490	978	45	1449	1.48	1.25	0.096	0.158
Slashdot	89777116	3782	22	1079	0.29	1.18	0.041	0.139
Enron	90296206	1702	53	1001	0.59	3.38	0.442	0.096

[1] Plain corpus; stimulus modality labels.

[2] Ontology annotated corpus; cognitive paradigm class labels.

One important feature of the data sets analyzed here is that they are unusually small (in terms of instances) and large (in terms of features) compared to many other standard data sets (compare d and N columns, also presented as a ratio in the d/N column). We expect in ongoing research to make use of larger pools of data from BrainMap which lead to complexities greater than 1×10^8 or an order of magnitude larger than the standard test sets in **Table 4**. If the dictionaries do not dramatically expand, this leads to d/N ratios closer to 1. Note that there are test sets in use, such as the MEDLINE baseline distributions (`www.nlm.nih.gov/bsd/licensee/baseline.html`) or OHSUMED (ir.ohsu.edu/ohsumed/ohsumed.html), among others, that are comparable with or exceed these larger sizes. However the data sets derived from the scientific literature will continue to have a particularly rich text feature space and therefore large d values.

Focusing on aspect of the data, the number of features is ≥ 3603 for all three corpora, and only 247 instances. The ability to identify synonyms or reduce this d through other means may improve performance, which is the scope of future work. The Colorado Richly Annotated Full Text Corpus (CRAFT; bionlp-corpora.sourceforge.net/CRAFT/index.shtml) is a counter-example, including only 67 papers originally, but that includes full text, and a substantial effort at detailed syntactic annotation and concept identification, with a final count of 793,627 tokens and many thousand annotations (Bada et al., 2012; Verspoor et al., 2012). Their annotations were focused on syntactic parsing of example genetic literature, and as such, the annotations were parts of speech and similar tags, rather than our goal of identifying multiple labels from different possible dimensions specific to neuroimaging experiments. Their parsing results are promising, however, for future more sophisticated applications to this domain of biomedical literature text and concept mining.

Note also that our data sets have labels from specific non-interchangeable dimensions; they are not simply a single bag of multi-label possibilities. Thus, as repeatedly noted above, they are not directly comparable to the common test cases. While the number of labels, LC_{avg}, and other measures are within the range used in other corpora, our data have relatively low complexity due to the small number of instances (247), an order of magnitude less than most other data sets used in this work. See Madjarov et al. (2012) and Read et al. (2011) for summary statistics on several additional comparable data sets.

A second issue is of course the use of abstracts only, rather than full text, in this work. As noted in Cohen et al. (2010), the linguistic content of abstracts is different from the content and structure of the full text of the document. As full text documents which are annotated with standardized terms from CogPO or other ontologies for human neuroimaging experiments become more plentiful, it is expected that the use of the Methods sections from those papers will lead to better performance. However, at the moment there are no readily accessible collections of the methods, or other sections, of papers making direct experimentation impossible. As more full-text is curated, it may be possible to extract other sections of technical papers for analysis.

The structure of ontologies for biomedical annotation certainly requires some consideration. As noted in Bada and Hunter (2011), ontologies for full-text, generic biomedical annotation should meet a number of requirements. CogPO meets several of these requirements, being a mid-level ontology with defined terminology and built on BFO, but it falls short of having richly defined relationships, logically constrained definitions that are unambiguous, and its representation of synonyms and acceptable alternative terms is sorely lacking. There need to be many levels between specific terms (or synonym classes) and high level concepts that are very abstract; this allows for retrieving similar results or being able to generalize to related terms. This is an area that appears open to formal analysis, but to date this analysis is lacking.

These richly-defined relationships and definitions specified in formal logic are less relevant for the kinds of classifiers we implemented in this work; we are using the labels as standard terms without any of the logical constraints or relationships defined across ontological classes. The labels here are used more as a controlled vocabulary than as an ontology *per se*. But the ability to identify alternative forms (synonyms) of labels would certainly improve performance, as would having a deeper hierarchy, with general classes broken into subclasses. For example, identifying that "Auditory Oddball" and "Spatial Oddball" are both "Oddball" paradigm classes, would allow the label "Oddball" to be identified without being completely correct, as a generalization of the finest-grained correct label. Incorporating this level of performance as a recommended term could facilitate the human annotator's job, as they now have a good reason to believe the Paradigm Class is an Oddball and only need to consider a more limited number of subclasses as potential annotations. It is worth mentioning that this conditionalization can be exploited by machine learning algorithms (Jones et al., 2013).

While machine-learning and text mining techniques have been applied in various biomedical domains to facilitate annotation or tagging, applications to human neuroimaging are rare, and the application to replicating expert-provided annotations regarding cognitive experimental details is available only through databases such as BrainMap or the similar Brede database (neuro.imm.dtu.dk/services/brededatabase/). The Neurosynth project (www.neurosynth.org) is an innovative text-mining effort on full-text versions of many neuroimaging papers, tagging papers and their imaging results with the most common words identified in the text; this allows users to identify what papers which have identified activations in a particular brain region have been written about. But to date it has focused on repetition of words for tagging, rather than identifying what the details of the experiments are, and thus what the results of the experiment might indicate. Given the motivating problem of facilitating curation—automatically identifying the appropriate annotations for a neuroimaging experiment—the performance of fairly basic classifiers indicates that some of the annotations can be identified quite accurately using these methods. Using a combination of binary relevance and naive Bayes can give a fairly good guess for the response type, and the response and stimulus modalities. The classifications for other dimensions using these methods would have to be considered suggestions to be confirmed, denied, or added to based on the human expert's judgment.

DISCLOSURE/CONFLICT-OF-INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

ACKNOWLEDGEMENT

The authors wish to thank Dr. Jesse Read for both the development of the MEKA software and for his assistance via the MEKA mailing list.

Funding: This project was supported by a grant from NIMH/NIH (1R56MH097870-01) to Drs. Jessica Turner and Angela Laird.

REFERENCES

Bada, M., M. Eckert, D. Evans, K. Garcia, K. Shipley, D. Sitnikov, W. A. Baumgartner, K. B. Cohen, K. Verspoor, J. A. Blake, et al. (2012). Concept annotation in the CRAFT corpus. *BMC bioinformatics 13*(1), 161.

Bada, M. and L. Hunter (2011). Desiderata for ontologies to be used in semantic annotation of biomedical documents. *Journal of Biomedical Informatics 44*(1), 94–101.

Bird, S., E. Klein, and E. Loper (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Sebastopol, CA: O'Reilly.

Bird, S. and E. Loper (2004). NLTK: The Natural Language Toolkit. In *Proceedings of the ACL demonstration session*, Barcelona, pp. 214–217. Association for Computational Linguistics.

Bug, W. J., G. A. Ascoli, J. S. Grethe, A. Gupta, C. Fennema-Notestine, A. R. Laird, S. D. Larson, D. Rubin, G. M. Shepherd, J. A. Turner, et al. (2008). The NIFSTD and BIRNLex vocabularies: building comprehensive ontologies for neuroscience. *Neuroinformatics 6*(3), 175–194.

Bzdok, D., A. R. Laird, K. Zilles, P. T. Fox, and S. B. Eickhoff (2012). An investigation of the structural, connectional, and functional subspecialization in the human amygdala. *Human Brain Mapping*.

Cherman, E. A., M. C. Monard, and J. Metz (2011). Multi-label Problem Transformation Methods: a Case Study. *CLEI Electron. J. 14*(1).

Cohen, A. M. and W. R. Hersh (2005). A survey of current work in biomedical text mining. *Briefings in Bioinformatics 6*(1), 57–71.

Cohen, K. B., H. Johnson, K. Verspoor, C. Roeder, and L. Hunter (2010). The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC bioinformatics 11*(1), 492.

Eyheramendy, S., D. D. Lewis, and D. Madigan (2003). On the Naive Bayes Model for Text Categorization.

Fan, R.-E. and C.-J. Lin (2007). A study on threshold selection for multi-label classification. Technical report, Department of Computer Science, National Taiwan University.

Farrell, M. J., A. R. Laird, and G. F. Egan (2005). Brain activity associated with painfully hot stimuli applied to the upper limb: A meta-analysis. *Human Brain Mapping 25*(1), 129–139.

Fitzgerald, P. B., T. J. Oxley, A. R. Laird, J. Kulkarni, G. F. Egan, and Z. J. Daskalakis (2006). An analysis of functional neuroimaging studies of dorsolateral prefrontal cortical activity in depression. *Psychiatry Research-Neuroimaging Section 148*(1), 33–46.

Fox, P. T., A. R. Laird, S. P. Fox, P. M. Fox, A. M. Uecker, M. Crank, S. F. Koenig, and J. L. Lancaster (2005). BrainMap taxonomy of experimental design: description and evaluation. *Human brain mapping 25*(1), 185–198.

Fox, P. T. and J. L. Lancaster (2002). Mapping context and content: the BrainMap model. *Nature Reviews Neuroscience 3*(4), 319–321.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter 11*(1), 10–18.

Howe, D., M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, S. S. Pierre, S. Twigger, O. White, and S. Y. Rhee (2008). Big data: The future of biocuration. *Nature 455*(7209), 47–50.

Jones, T., C. Chakrabarti, J. Xu, M. D. Turner, G. F. Luger, A. Laird, and J. A. Turner (2013). Modeling ontology-based annotation processes for neuroimaging abstracts using a stochastic framework. *Annual Meeting of the Organization for Human Brain Mapping*.

Laird, A. R., S. B. Eickhoff, K. Li, D. A. Robin, D. C. Glahn, and P. T. Fox (2009). Investigating the functional heterogeneity of the default mode network using coordinate-based meta-analytic modeling. *The Journal of Neuroscience 29*(46), 14496–14505.

Laird, A. R., P. M. Fox, C. J. Price, D. C. Glahn, A. M. Uecker, J. L. Lancaster, P. E. Turkeltaub, P. Kochunov, and P. T. Fox (2005). ALE meta-analysis: Controlling the false discovery rate and performing statistical contrasts. *Human brain mapping 25*(1), 155–164.

Laird, A. R., J. J. Lancaster, and P. T. Fox (2005). Brainmap. *Neuroinformatics 3*(1), 65–77.

Lancaster, J. L., A. R. Laird, P. M. Fox, D. E. Glahn, and P. T. Fox (2005). Automated analysis of meta-analysis networks. *Human brain mapping 25*(1), 174–184.

Langlotz, C. P. (2006). RadLex: A New Method for Indexing Online Educational Materials1. *Radiographics 26*(6), 1595–1597.

Lok, C. (2010). Speed reading. *Nature 463*, 28.

Madjarov, G., D. Kocev, D. Gjorgjevikj, and S. Džeroski (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition 45*(9), 3084–3104.

McCallum, A., K. Nigam, et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, Volume 752, pp. 41–48. Citeseer.

Menzies, L., S. R. Chamberlain, A. R. Laird, S. M. Thelen, B. J. Sahakian, and E. T. Bullmore (2008). Integrating evidence from neuroimaging and neuropsychological studies of obsessive-compulsive disorder: the orbitofronto-striatal model revisited. *Neuroscience & Biobehavioral Reviews 32*(3), 525–549.

Modi, H. and M. Panchal (2012, December). Experimental Comparison of Different Problem Transformation Methods for Multi-Label Classification using MEKA. *International Journal of Computer Applications 59*(15), 10–15.

Platt, J. (1998). Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In B. Schlkopf, C. Burges, and A. Smola (Eds.), *Advances in Kernel Methods – Support Vector Learning*, pp. 42–65. Cambridge, MA: MIT Press.

Poldrack, R. A., A. Kittur, D. Kalar, E. Miller, C. Seppa, Y. Gil, D. S. Parker, F. W. Sabb, and R. M. Bilder (2011). The cognitive atlas: toward a knowledge foundation for cognitive neuroscience. *Frontiers in neuroinformatics 5*.

Read, J., B. Pfahringer, and G. Holmes (2008). Multi-label Classification Using Ensembles of Pruned Sets. In *ICDM*, pp. 995–1000. IEEE Computer Society.

Read, J., B. Pfahringer, G. Holmes, and E. Frank (2009). Classifier Chains for Multi-label Classification. In *Proc 13th European Conference on Principles and Practice of Knowledge Discovery in Databases and 20th European Conference on Machine Learning*, Bled, Slovenia. Springer.

Read, J., B. Pfahringer, G. Holmes, and E. Frank (2011). Classifier chains for multi-label classification. *Machine Learning 85*, 333–359.

Rennie, J. D., L. Shih, J. Teevan, and D. Karger (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Volume 20, pp. 616.

Rosse, C. and J. L. V. Mejino (2003). A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics 36*, 478–500.

Santos, A., A. Canuto, and A. Neto (2011). A comparative analysis of classification methods to multi-label tasks in different application domains. *International journal of computer Information systems and Industrial Management Applications". ISSN*, 2150–7988.

Spyromitros, E., G. Tsoumakas, and I. Vlahavas (2008). An empirical study of lazy multilabel classification algorithms. In *Artificial Intelligence: Theories, Models and Applications*, pp. 401–406. Springer.

Trieschnigg, D. (2010). Proof of concept: concept-based biomedical information retrieval. *SIGIR Forum 44*(2), 89.

Trieschnigg, D., P. Pezik, V. Lee, F. de Jong, W. Kraaij, and D. Rebholz-Schuhmann (2009). MeSH Up: effective MeSH text classification for improved document retrieval. *Bioinformatics 25*(11), 1412–1418.

Tsoumakas, G. and I. Katakis (2007). Multi Label Classification: An Overview. *International Journal of Data Warehousing and Mining 3*(3), 1–13.

Tsoumakas, G., I. Katakis, and I. P. Vlahavas (2010). Mining multi-label data. *Data mining and knowledge discovery handbook*, 667–685.

Tsoumakas, G., I. Katakis, and I. P. Vlahavas (2011). Random k-Labelsets for Multilabel Classification. *IEEE Trans. Knowl. Data Eng. 23*(7), 1079–1089.

Turner, J. A. and A. R. Laird (2012). The Cognitive Paradigm Ontology: Design and Application. *Neuroinformatics 10*(1), 57–66.

Verspoor, K., K. Cohen, A. Lanfranchi, C. Warner, H. Johnson, C. Roeder, J. Choi, C. Funk, Y. Malenkiy, M. Eckert, et al. (2012). A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC bioinformatics 13*(1), 207.

Witten, I. H., E. Frank, and M. A. Hall (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3 ed.). Amsterdam: Morgan Kaufmann.

Zhang, H. (2004). The Optimality of Naive Bayes. In V. Barr and Z. Markov (Eds.), *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*. AAAI Press.

Zhang, H. (2005). Exploring Conditions For The Optimality Of Nave Bayes. *IJPRAI 19*(2), 183–198.

Zhang, M.-L. and Z.-H. Zhou (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition 40*(7), 2038–2048.