Automated Annotation of Neuroimaging Abstracts for Cognitive Experiments

*Chayan Chakrabarti [1], George F. Luger [1], Angie R. Laird [2], and Jessica A. Turner [3]**

[1] Department of Computer Science, University of New Mexico, Albuquerque, NM. [2] Department of Radiology, University of Texas Health Sciences Center, San Antonio, TX. [3] Mind Research Network, Albuquerque, NM.

ABSTRACT

Motivation: This work explores automatic annotation of fMRI studies based on standard terms from the Cognitive Paradigm Ontology (CogPO). We have implemented an initial text mining approach on a subset of texts of abstracts from the BrainMap database (www.brainmap.org), to automate the expert annotations from the BrainMap schema and CogPO terms. We measured the performance of a basic K-nearest-neighbor (KNN) approach on the title and abstract text of the corpus, in predicting the correct annotations. The results are better than chance, which is promising given the high-dimensional nature of the problem. We also experiment with n-gram models. Our work points toward the use of semantic models more complex than simple distance among abstracts.

1 INTRODUCTION

One of the largest databases of neuroimaging results in humans is the BrainMap database (www.brainmap.org). Since the early 1990s, its curators have been manually extracting descriptions of first PET and then fMRI experiments, and storing each paper's results in a standardized system for ease of retrieval (Fox et al 2005, Laird et al 2005). The BrainMap software suite provides multiple applications that interface with the database to submit papers for entry, search, retrieve, and filter studies, and to perform quantitative meta-analysis. This system has facilitated reviews and meta-analyses of the literature through identifying consistent subsets of experiments (Laird et al. 2005). The ability to perform meta-analyses to identify replicated results is part of the toolset needed to explore the different cognitive constructs underlying brain function in various disorders, such as the constellation of schizophrenia, bipolar disorder, depression, and autism.

The ability to run large-scale meta-analyses demands the ability to easily identify studies using the same (or similar enough) experimental methods and subjects. The BrainMap method for describing experiments has evolved into a taxonomy composed chiefly of structured keywords that categorize the experimental question addressed, the imaging methods used, the behavioral conditions during which imaging was acquired, and the statistical contrasts performed. The schema that BrainMap uses to describe experiments has been used to form the backbone of the Cognitive Paradigm Ontology. That ontology (Turner & Laird 2012) uses the keywords from BrainMap and explicitly represents the implicit definitions and relationships among them. The driving force behind CogPO's design is to allow published experiments implementing similar behavioral task characteristics to be linked, despite the use of alternate vocabularies. CogPO has been submitted to the National Center for Biomedical Ontologies (NCBO) Bioportal, and is available for use in semantic annotation and reasoning.

While the value of the BrainMap project has been proven, the number of publications in the literature far outweighs the number of publications that have been included in the database. The human step of reading the paper and determining its annotations is currently a bottleneck. In this project, we aim to find a method for replacing the human step with automated suggestions for the experimental paradigm terms.

2 BACKGROUND

A variety of methods have already been developed for automated annotations of free text within the biomedical research community. The NCBO Annotator, for example, will take free text and use efficient concept-recognition techniques to suggest annotations from the BioPortal repository of ontologies (Shah et al 2009). The Neuroscience Information Framework (Gardner et al 2008) uses ontological annotations of a broad variety of neuroscience resources to retrieve information for user queries. Neither of these, however, have broadly attempted to recreate a human expert's annotations on a curated dataset in their application of ontological terms.

The CogPO ontology considers experiments to have experimental conditions; experimental conditions are combinations of stimulus, response, and instructions (Turner & Laird, 2012). Each of these classes has a fixed number of terms, and the papers in the BrainMap database have all been annotated based on the experimental conditions in the experiments, and the stimulus types, response types, and instructions used in each case. This provides a gold standard for developing an annotation algorithm.

* To whom correspondence should be addressed.

3 METHODS

We experiment with two categories of methods: methods emphasizing presence of high-entropy words, and methods emphasizing the sequence in which the words occur. High-entropy words are those, which add more discriminating information. These are likely to be technical terms relevant to the domain. In the second category, we examine the sequence in which certain words tend to occur in the corpus, rather than the words themselves. Both methods are described below.

3.1 Experimental Setup 1.

Our corpus of documents contained 327 published papers from the BainMap Database. Each paper had an associated stimulant, response to stimulant, and instructions, all of which were annotated. Papers were restricted to the subfield of fMRI attention studies. The full text of the abstract and title were then downloaded from PubMed using EUtils, for the citations, which had them. This resulted in a final corpus of 327 abstracts and titles, each of which had a minimum of one and a maximum of four annotations per Stimulus, Response, and Instruction. There were twenty-seven different stimulus terms used in these annotations.

The next step was to create a dictionary that would represent all the words in the corpus of papers. We wanted to keep the most discriminative words in the dictionary, but at the same time keep the dictionary representative of the words in the corpus. To ensure this, we removed all stop words from the dictionary. For example, words like because, this, is, was, when, etc. were removed since they are not very discriminative. We used a standard English stop word list (http://www.ranks.nl/resources/stopwords.html). We also wanted make sure that all the words in the dictionary were either legitimate English-language words, or were part of standard medical terminology and specifically, neuroscience terminology. We used a list of words from a standard English dictionary, and a list of words from a medical dictionary. We constructed our dictionary from the set of all words contained in the corpus of paper abstracts, that were not a part of the list of stop words, and were part of the standard and medical dictionaries.

To ensure that we did not have redundancies in the dictionary with many forms of the same roots, our next step was to stem all the words in the dictionary and the corpus. In linguistic morphology and information retrieval, stemming is the process for reducing inflected or sometimes, derived words to their stem, base or root form, generally a written word form. The stem need not be identical to the morphological root of the word, it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. The Porter stemming algorithm (or 'Porter stemmer') is a process for removing the commoner morphological and in flexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems. (Porter 1980) We used the porter stemmer to reduce both the dictionary and the corpus to the root stems of existing words, thereby reducing redundancy.

Now we had a dictionary with 2,241 words in it. We then converted each abstract in the corpus to a 2,241 element vector, in which each element represents the count of the number of words from the dictionary that are present in the abstract. Thus our entire corpus is new represent by 327 vectors of 2,241 elements each. We can now visualize each abstract as a point in a 2,241-dimensional space. We consider the Euclidean distance between the point-abstracts as a proxy for the similarity between the contents of the abstract. Our hypothesis is that abstracts that are tightly bunched together in this space will be similar in content, and hence more likely to be annotated with the same terms.

Table 1: Some statistical metrics on the space of point-abstracts. The stimulus types above the double lines appear in the annotations more than 15 times, while those below appear less than 15 times. Other stimulus types appear very few times, and have not been included here.

Stimulus Type	Mean Pairwise Distance	Standard Deviation
All	9.2491	1.1856
Words	8.2597	1.9254
Shapes	8.8745	1.7743
Pictures	11.1896	2.0172
Letters	9.3124	1.3449
Digits	9.0212	1.7234
Faces	10.0132	2.1134
Fixation	9.7427	1.9847
Symbols	9.5926	2.0031

We used the K-nearest neighbor to determine annotations for each point-abstract in relation to its proximity to other point-abstracts. K-nearest neighbor is a good fit for our problem since it is a non-parametric lazy learning algorithm. It does not make any assumptions on the underlying data distribution, and in our problem we do not know the distribution of the point-abstracts in advance. Since it is a lazy learning algorithm, we do not need an explicit training phase. (Duda & Hart, 2001)

We split our data set of point-abstracts in to a training and generalization set in a 1:2 ratio, i.e., there were 109 points abstracts in the training set, and 218 point-abstracts in the generalization set. This split was done at random; for comparison, we also performed a leave-one-out analysis. The point-abstracts in the training set were annotated with the stimulus types associated with that corresponding paper.

Table 2: Pairwise distance between the centroids of the 5-most common clusters.

	Words	Shapes	Pictures	Letters	Digits
Words	0	12.35	13.64	7.42	8.93
Shapes	12.35	0	13.41	12.94	13.05
Pictures	13.64	13.41	0	14.17	14.28
Letters	7.42	12.94	14.17	0	13.91
Digits	8.93	13.05	14.28	13.91	0

We then apply the K-nearest neighbor algorithm to the entire space of point-abstracts to automatically annotate the point-abstracts in the generalization set. A point-abstract in the generalization set is annotated with the stimulus type as determined by the weighted proximity of its K-nearest neighbors in the space.

Table 3: Results of K-nearest neighbor algorithm for automatic annotation of point-abstracts using 2-fold cross-validation.

Value of K	Annotation Accuracy
1	29.67%
5	43.61%
10	52.94%
20	53.11%
50	47.22%
100	30.06%

We repeated this experiment for K = 1, K = 5, K = 10, K= 20, K = 50, and K = 100. For each value of K, the experiment was run 10,000 times, each time a new training and generalization set was selected at random. The results of the generalization were compared with actual annotations, and were averaged over the 10,000 runs.

3.2 Results

As we can see from Tables 1 and 2, a simple statistical check does not indicate any clearly discernible difference in within-cluster distances across stimulus types, or any extremes of distance between individual clusters in the point-abstract space. The stimulus types words, shapes, pictures, letters, and digits are the most occurring annotations in the abstract corpus, each occurring more than 15 times. Other annotations like faces, fixation, and symbols occur fewer than 15 times. The remaining annotations occur only sparingly in the abstract corpus, fewer than 5 times.

The accuracy of the K-nearest neighbor algorithm is shown in Tables 3 and 4. For K = 1, the point-abstract in the generalization set is just annotated with the same stimulus type of the point-abstract closest to it. For K = 5, the point-abstract in the generalization set is annotated with the most common stimulus type among the five nearest neighbors from the training set. This is repeated for values K = 10, K = 20, K = 50, and K = 100. We can observe a very similar trend for the leave-one-out cross validation results in Table 4.

Table 4: Results of K-nearest neighbor algorithm for automatic annotation of point-abstracts using leave-one-out cross-validation.

Value of K	Annotation Accuracy
1	26.33%
5	41.92%
10	51.69%
20	49.73%
50	48.39%
100	27.34%

We can discern some obvious trends from the results. The results are better than random guessing. Since most of the abstracts in the corpus are annotated with one of the five stimulus types mentioned in Table 1, a random guess would yield around 20% accuracy. In our experiments, we were getting accuracies far above that even in the worst cases. As the value of K increases from K = 1 to K = 10, so does the accuracy. This is because as we increase the number of neighbors influencing the point-abstract in the training set, the weighting can take in to consideration more information about similar point-abstracts, which are likely to be in close proximity. However as the number of neighbors increases further, for values K = 20 to K = 100, the accuracy decreases. This is because as the size of the neighborhood approaches the size of the entire training set (K = 109), local information regarding proximity is lost and this decreases the effect of the weighting of the nearest neighbors.

3.3 Experimental Setup 2.

In this experiment, we consider sequences of words, rather than the words themselves. In this model, the probability of a sequence of words occurring is modeled using a unigram (Luger 2008). Consider a sequence of words w_0, w_1, w_2, …. w_n. We can assume that these words occur independent of each other. Hence, we can model the joint probability of the sequence of words as.

$$P(w_1, w_2, \ldots. w_n) = P(w_1)P(w_2|w_1)P(w_3|w_2)\ldots.P(w_n|w_{n-1})$$

Thus, we can now encode every abstract as a n by n matrix, where each cell (i,j) of the matrix, represents $P(w_i|w_j)$. We restrict our analysis to unigrams due to computational expense. We can now compare the probability distribution matrices of each encoded abstract using KL-divergence (Luger 2008).

3.4 Results

However, we get very poor results form this technique. The probabilities were uniformly very close to zero, leading to unusable KL-divergence measures. Unigram models typically are used to represent spoken language models. Since abstracts, by practice, are written in a succinct fashion, unigrams do not model them well.

4 DISCUSSION

We have explored some basic approaches to automated annotation techniques, using a gold-standard corpus of neuroimaging abstracts. We find that the initial KNN models are promising, but n-gram models will require either a much larger corpus or more of the text than just the abstract.

One challenge was the curse of dimensionality. We were essentially working in a 2,241 dimensional space with just 327 data points. A larger set of data points could alleviate this problem. But a better solution is to reduce the dimensionality of the space itself. Our technique of reducing abstracts to word vectors, which had the same cardinality of the dictionary, only took in to account the number of occurrences of key discriminative words n the abstracts. The next step is to take in to account more sophisticated characteristics of the abstracts. Our future work will representing the abstracts using textual and language models that leverage the semantic artifacts of the contents of the abstracts (Trieschnigg et al. 1999). This will take in to account richer linguistic features like acronyms, synonyms, etc., and also semantic features like concepts expressed.

In most real world annotation problems including this one, most entities have several annotations—a paper can describe several experiments, each with several conditions and multiple stimulus types, for example. The number of annotations the algorithm got right, or the percentage of annotations it accurate predicted for each entity considered separately, could provide very different performance metrics. Ideas from folksonomy research community, who semantically mine pictures using tags for information, may be relevant in this case.

We expect that even if automated annotation techniques cannot completely replace human annotators, they may be designed to work in complement with human annotators. We can envision such techniques being able to guide human annotators in the right direction by capturing high level semantics of the corpus, and identifying a high-probability subset of terms; in the BrainMap case, this would provide an initial filtering and partial annotation of the papers, speeding up the process of entry.

REFERENCES

Duda & Hart. *Pattern Classification*. (2001) 2nd Edition. Wiley-Interscience.

Eickhoff SB, Laird AR, Grefkes C, Wang LE, Zilles K, Fox PT. (2009) *Coordinate-based activation likelihood estimation meta-analysis of neuroimaging data: a random-effects approach based on empirical estimates of spatial uncertainty.* Hum Brain Mapp. 30(9):2907-26. PMCID: PMC2872071.

Fox PT, Laird AR, Fox SP, Fox PM, Uecker AM, Crank M, Koenig SF, Lancaster JL. (2005) *BrainMap taxonomy of experimental design: description and evaluation.* Hum Brain Mapp. 25(1):185-98.

Gardner D, Goldberg DH, Grafstein B, Robert A, Gardner EP. (2008) *Terminology for Neuroscience Data Discovery: Multi-tree Syntax and Investigator-Derived Semantics.* Neuroinformatics. 6(3):161-74. PMCID: PMC2655120.

Glahn DC, Ragland JD, Abramoff A, Barrett J, Laird AR, Bearden CE, Velligan DI. (2005) *Beyond hypofrontality: a quantitative meta-analysis of functional neuroimaging studies of working memory in schizophrenia.* Hum Brain Mapp. 25(1):60-9.

Gupta A, Bug W, Marenco L, Qian X, Condit C, Rangarajan A, Müller HM, Miller PL, Sanders B, Grethe JS, Astakhov V, Shepherd G, Sternberg PW, Martone ME. *Federated Access to Heterogeneous Information Resources in the Neuroscience Information Framework (NIF).* Neuroinformatics. 2008;6(3):205-17. PMCID: PMC2664632.

Laird AR, Lancaster JL, Fox PT. (2005*) BrainMap: the social evolution of a human brain mapping database.* Neuroinformatics. 3(1):65-78.

Laird AR, Eickhoff SB, Kurth F, Fox PM, Uecker AM, Turner JA, Robinson JL, Lancaster JL, Fox PT. (2009) *ALE Meta-Analysis Workflows Via the Brainmap Database: Progress Towards A Probabilistic Functional Brain Atlas.* Front Neuroinformatics. 3:23. PMCID: PMC2715269.

Luger G. (2008) *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Addison Wesley; 6th Edition.

Marenco L, Li Y, Martone ME, Sternberg PW, Shepherd GM, Miller PL. (2008) *Issues in the Design of a Pilot Concept-Based Query Interface for the Neuroinformatics Information Framework. Neuroinformatics.* 6(3):229-39. PMCID: PMC2663521.

Porter, M.F. (1980) *An algorithm for suffix stripping, Program,* 14(3) :130-137.

Shah NH, Bhatia N, Jonquet C, Rubin D, Chiang AP, Musen MA. (2009) *Comparison of concept recognizers for building the Open Biomedical Annotator.* BMC Bioinformatics. 10 Suppl 9:S14. PMCID: PMC2745685.

Shah NH, Jonquet C, Chiang AP, Butte AJ, Chen R, Musen MA. (2009) *Ontology-driven indexing of public datasets for translational bioinformatics.* BMC Bioinformatics. 10 Suppl 2:S1. PMCID: PMC2646250.

Trieschnigg D, Pezik P, Lee V, de Jong F, Kraaij W, Rebholz-Schuhmann D. (2009) *MeSH Up: effective MeSH text classification for improved document retrieval.* Bioinformatics. 25(11):1412-8. PMCID: PMC2682526.

Turner JA, Laird AR. (2012) *The cognitive paradigm ontology: design and application.* Neuroinformatics. 10(1):57-66.