sciendo

Case Study **Open Access**

Will R. Thomas*, Benjamin Galewsky, Sandeep Puthanveetil Satheesan, Gregory Jansen, Richard Marciano, Shannon Bradley, Jong Lee, Luigi Marini, Kenton McHenry

Petabytes in Practice: Working with Collections as Data at Scale

https://doi.org/10.2478/dim-2019-0004
received August 2, 2018; accepted March 1, 2019.

Abstract: The emerging transdiscipline of Computational Archival Science (CAS) links frameworks such as Brown Dog and repository software such as Digital Repository At Scale To Invite Computation (DRAS-TIC) to yield an understanding of working with digital collections at scale for cultural data. The DRAS-TIC and Brown Dog projects here serve as the basis for an expandable distributed storage/service architecture with on-demand, horizontally scalable integrated digital preservation and analysis services.

Keywords: JCDL workshop proceedings, data repositories, parallel processing, machine learning

1 Introduction

In this paper, we design a system for incremental interactive learning of an annotated publishable corpus derived from an archival collection. Learning here is a task for both the archival institution and community organized around the archival collection and a machine learning (ML) pipeline which refines the annotation of the corpus.

Digitizing a subset of the paper records of the George Meany Memorial Archive hosted at the University of Maryland created images from boxes of the American Federation of Labor—Congress of Industrial Organizations (AFL-CIO) archived records central to organized labor's participation in the Civil Rights Movement. Our task is to define a means to build an annotated corpus based on those images which can expand to incorporate additional images created through subsequent institutional and community digitization efforts and refinements of the model. In so doing, we see how Computational Archival Science (Marciano et al., 2018) can help theorize new relationships between communities and the memory institutions that serve them. We can see how we can bring computing to the data in a record in order to build deeper descriptions of it than would otherwise be possible, as well as facilitating recontextualization and incremental improvement in record description, and how we can scale storage horizontally as the number of images increases.

This paper is organized as follows. We review the literature to identify methods which can be placed in a pipeline to learn a corpus from images. We then describe how such a pipeline can be realized by building on infrastructure components from the Brown Dog and DRAS-TIC projects (these two projects are described in more detail below).

2 Background

We must weave together a number of concepts and methods from the literature to incrementally build an annotated corpus from archival records. These concepts are as follows:

- treebank corpus annotation
- relational lenses
- document interpretation acts
- weak supervised learning
- convolutional neural networks
- long short-term memory
- bidirectional long short-term memory
- multimodal long short-term memory
- connectionist temporal classifier

***Corresponding author: Will R. Thomas,** University of Maryland, College Park, MD, United States. Email: wthomas4@umd.edu
Gregory Jansen, Richard Marciano: University of Maryland, College Park, MD, United States
Benjamin Galewsky, Sandeep Puthanveetil Satheesan, Shannon Bradley, Jong Lee, Luigi Marini, Kenton McHenry: University of Illinois at Urbana–Champaign, Urbana, IL, United States

We can start with treebank corpus annotation itself, modeled after the Penn Treebank Corpus (Marcus, Santorini, & Marcinkiewicz, 1993). The Penn Treebank Corpus was built using an iterative process of classifying text by syntax ("tagging" words) and then creating syntax trees to relate words to a sentence-level syntactic structure ("chunking" or "bracketing"). Labels were initially assigned algorithmically and then refined through a hybrid process of manual label editing and revised algorithmic models (Marcus, Santorini, & Marcinkiewicz, 1993).

Relational lenses (Bohannon, Pierce, & Vaughan, 2006) are the next conceptual touchpoint. A relational lens is a view in a relational database so defined that updates to the underlying base relational variables in the view have their effects as clearly defined as does the query language defining the view itself; that is to say, the view is defined bidirectionally (Bohannon, Pierce, & Vaughan, 2006). The labels for syntax in a corpus can be viewed through such a relational lens; therefore, they can be updated through a relational data manipulation language.

Document interpretation acts (Bradley & Pasin, 2013) capture the event of creating and updating of data through such a relational lens in an ontology describing documents in relation to the people and places in them. It links these various prosopographic entities such as named persons discoverable in text to the means of finding them.

Weak Supervised Learning (Craven et. al., 1999) is a hybrid iterated machine-learning approach where document interpretation acts including both manually and algorithmically generated labels for testing and training data are combined to construct datasets for model training larger than those which would be available with manual labels alone.

Convolutional Neural Networks (CNNs) (LeCun, Kavukcuoglu, & Farabet, 2010) are modeled after mammalian vision, where invariant features of the visual environment that mark a significant event (such as the movement of a predator or prey across the field of vision) are recognized amid the range of incoming stimuli. Its function is to take a first pass at extracting features. It is attempting to recognize characters of text by recognizing the invariant features characteristic of letters in the words.

Long Short Term Memory (Hochreiter & Schmidhuber 1997) was a solution to feedback/feedforward degrading to zero or infinity in neural network recognizers. By embedding the ability for signals to be remembered during processing, signals encountered during different intervals of processing can be processed together while mitigating the loss of information across the time span between them.

Bidirectional Long Short Term Memory (BiLSTM) (Graves, Fernández, & Schmidhuber, 2006) improves on the ability of LSTMs to connect arbitrarily temporally distant features by removing directional constraints on that connection. The context for any one feature need not be immediately adjacent to be recognized, but now can be features seen later contextualizing earlier seen features as well as vice versa. Here, this means that newly recognized words, tags, or relationships can provide guidance to resolve ambiguities in ones previously seen.

Multimodal Long Short Term Memory (Ren et. al., 2016) allows training across multiple recognizers to connect stimuli received at the same time and reinforce learned connections between them, so that a text recognizer and a relationship extractor can for example provide feedback to one another during training. As we will use this concept, we extend multimodal deep learning (Ngiam et. al., 2011) by considering a pair of recognizers which operate on simultaneous parallel inputs where both of those inputs are the same.

Connectionist Temporal Classifiers (CTCs) (Graves et. al., 2006) recognize the significance of an apparently insignificant interval by seeing that it is actually an interruption in a sequence as opposed to an empty space between sequences. It changes the loss function to classify these interruptions properly. In so doing, it maximizes the probability of recognizing that a sequence which has been broken into segments (extraneous spaces introduced by defects in imaging or hyphenization) is a unity.

Extant systems combine the CNN, multimodal BiLSTM, and CTC concepts into neural networks for speech recognition (Amodei et. al., 2015). In a speech recognition task, a word recognized at the current time can change the context of words already recognized, possibly to the point it changes the net's understanding of what word was recognized. An extant system producing text representations from images was created by Dropbox to handle smartphone digitization and OCR of business records such as receipts (Neuberg, 2017). Applying this net to a text-processing task takes advantage of the analogy between hearing speech and reading text (Mattingly, 1972) in that both are attention-driven tasks with possibly long delays between events that change the understanding or internal representation of perceptions. It also takes advantage of the externalization of memory in writing; memory of an acoustic event (outside of phenomena such as an echo) is all one has to refer to if one is connecting something heard now to something heard previously, whereas for reading one can recreate the previous perception of a word on demand alongside accessing one's memory of it.

3 Case Study

We consider the feasibility of extending the digitization capability of an archival institution to enable community researchers to be part of it. Community researchers with access to this capability would be able to image records utilizing their smartphone cameras and help build novel interpretations of those records as part of a corpus.

We envision researchers encountering records to be added to the corpus while examining the contents of a folder. They send the browser on their smartphone to a page hosted by the memory institution which lets them capture an image of the record, along with identifying information such as collection name, box and folder information, and any notes. The page automatically captures date/time and SHA256 fixity information for the image. When the researcher hits send, a system hosted by the institution goes into action. It extracts text from the image, cleans the text up, and publishes it as a text representation for the record image. Such a capacity allows for incremental production of an annotated corpus to happen regardless of current batch digitization efforts and can include records which are otherwise not prioritized for the digitization by the institution.

We intend to build on components from projects hosted at the University of Maryland and University of Illinois at Urbana–Champaign, respectively. Maryland is the host of the **Digital Repository at Scale that Invites Computation** (Jansen & Marciano, 2016) or DRAS-TIC stack to deploy data repositories that scale out to billions of files on potentially thousands of commodity servers. DRAS-TIC employs a Django content management system (CMS) front-end over a Cassandra (Lakshman & Malik, 2010) storage cluster to enable collections to scale horizontally. The core of the architecture is the US National Science Foundation (NSF)-funded **Brown Dog** project, which is a partnership between the University of Illinois Urbana-Champlain and the University of Maryland (McHenry et. al., 2017). The objective of Brown Dog was to create Digital Infrastructure Building Blocks (DIBBs) as modules to power next-generation digital collections. These modules are discoverable services that can operate within a framework capable of powering parallel pipelines for ingesting, transforming, preserving, and recomposing digital records and surrogates. This repository platform provides a web interface and standard data storage application programming interfaces (APIs) enabling a Brown Dog workflow to scale as the volume of ingested data and the size of the collection scale.

Brown Dog is a data transformation service for understanding unstructured data by means of auto-curation and indexing. It is built to help make sense of data that require access to a diverse set of software for processing. Brown Dog is a "super-mutt" of software that tries to leverage all available software tools toward auto-curation (Padhy et. al., 2015). It also encourages sharing of data transformation tools. The software tools are exposed as services and are available to its users to meet their data transformation needs. Brown Dog is now past its beta release and is heading toward the 1.0 release. Brown Dog currently supports a wide variety of use cases (Satheesan et. al., 2018), and Computational Archival Science nicely fits into its collection of use cases.

Brown Dog can handle complex data file conversions between formats and can orchestrate sequences of conversions and extractions. Internally, these microservices are implemented as containerized Clowder (Marini et. al., 2018) extractors. Clowder provides a metadata extraction bus and a way to package external software. Brown Dog manages the complexities of extractor dependencies and configuration, where an extractor proves computationally expensive. Brown Dog manages overall throughput handle scaling of these extractors by deploying them in a cluster which dynamically scales the number of active processes on demand.

We are proposing neural net extractors not in the Brown Dog catalog; implementing them will require us to address the Brown Dog Software Development Kit (SDK) for creating new extractors. Implementing code can focus on specific recognition or transformation tasks and leave to the SDK the connections to Brown Dog for data reading, metadata writing, and scaling.

The collection being targeted here is the George Meany Memorial Archive, the largest single donation of archival material to the University of Maryland (University of Maryland, 2016). Records in the archive comprise multiple record groups capturing documents, publications, imagery, media, and ephemera. Of specific interest are the records related to the AFL-CIO Civil Rights Division, which were assessed for digitization under the aegis of the African American Digital Humanities (AADHum) Initiative at the University of Maryland. With funding from Mellon, a team was able to perform folder-by-folder assessments of individual documents from multiple record groups, identifying both types of documents and entities named within those documents.

The significance of the Civil Rights Division records lies in their close documentation of organized labor's role in ending Jim Crow (the public and commercial legal structure of segregation and impunity for racist violence in the United States) as a partner of the Civil Rights Movement. The actual digitization is still in progress, so

record groups 1, 9, and 21 are the ones represented with surrogates. Materials within these record groups were picked owing to their box- and folder-level descriptions in the finding aid for the archive. A few hundred gigabytes of images of this material have currently been digitized.

In addition, this collection includes records in Record Group 20, which are records from the Information Department in the form of issues of the AFL-CIO News. This material was digitized by the University of Maryland and the digital surrogates produced are now hosted by the Internet Archive.

As per the permissions granted by the AFL-CIO, all of the records selected are publicly available for noncommercial use, so data for this project can be made publicly available. In order to reduce costs and make the implementation details open to the community as well, all the components in the architecture need to be open source.

4 Neural Net Training

The neural networks at the heart of the stack need to be trained in order to extract text well. Existing digitization efforts based on this collection provide a quantity of training data which will be needed for building and testing the outputs of the system. This existing training data come in two sets, a labeled image set and an unlabeled one.

The labeled image set consists of JPEG images produced by the University of Maryland with accompanying OCR text representations produced by the Internet Archive using ABBYY FineReader. These digital surrogates derive from records in Record Group 20, which are records from the Information Department in the form of issues of the AFL-CIO News. The OCR text representations are used as labels for supervised learning. In addition, the Penn Treebank-based NLTK (Loper & Bird, 2002) libraries will create labels for parts of speech (POS) tagging and chunking. All these labels will be stored in a materialized view in a PostgreSQL database attached to the Django CMS in DRAS-TIC.

The second set consists of JPEG images captured by digital cameras from the physical records in record groups 1, 9, and 21. Imaging was performed by a contractor not otherwise connected with this work. Initial metadata collected for the surrogates include originating record group, collection, box, and folder number for the underlying physical record, filename and path assigned at the time of capture, SHA256 fixity information, and format information captured by the identity program in the ImageMagick suite.

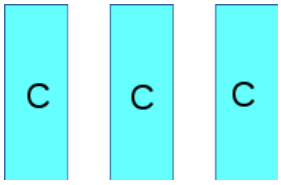

Figure 1. Storage Pool. C = Cassandra

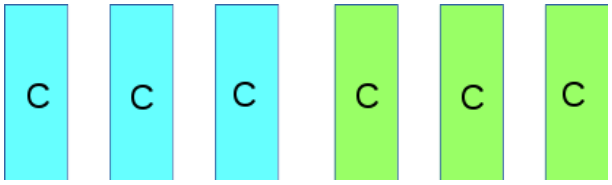

Figure 2. Expanded Storage Pool.

We rely on Apache Spark (Zaharia et. al., 2016) to distribute the training of the machine-learning models. Training and testing data are stored in Cassandra, as Spark integrates with it as a persistence layer. Although Spark has an ML library (Meng, et. al., 2016), the neural net models need to be implemented using a library which can realize neural nets. SparkNet (Moritz et. al., 2015) enables the use of frameworks which can implement a CNN or LSTM network, and of the neural net libraries it enables, and Caffe (Jia et. al., 2014) has a Python binding.

We use Docker images as abstractions for computing power. For our purposes, we can consider these images to at base be Debian stretch GNU/Linux, although Ubuntu, using the same package manager as Debian, would work similarly.

We assume our collection consists of a set of data in heterogeneous digital formats. We will assume a set of Docker containers running Cassandra and forming a cluster. This cluster holds our collection data. We can call it the storage pool.

Basing the storage pool on Cassandra means that we can scale the pool horizontally, increasing the size of storage it can manage by simply adding more nodes.

We can take advantage of this horizontal scaling to add an analytic capability operating over the storage pool. To do this, we augment the Docker image used to host Cassandra nodes for the storage pool. Each container based on this new image will run Cassandra and will be a part of the storage pool, but in addition, each container will have Spark installed and listen on Spark's port set so that it can join a Spark cluster. Just as we call a Cassandra cluster the storage pool, we can call a Spark cluster the analytic pool.

After installing Spark, we must install the SparkCassandra connector, so that a Spark node can

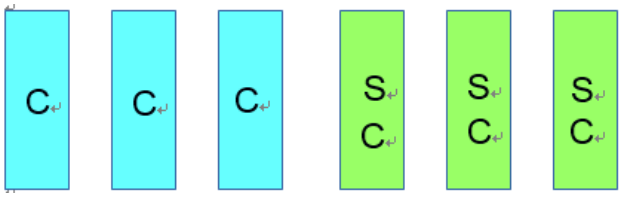

Figure 3. Storage and Analytic Pools. C = Cassandra, S = Spark

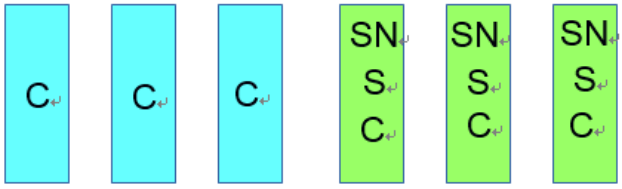

Figure 4. Storage and Analytic Pools with SparkNet. C = Cassandra, S = Spark, SN = SparkNet

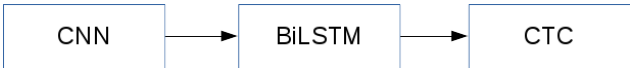

Figure 5. Neural Net Layers

access data from the storage pool through the Cassandra node in its container. We must also install SparkNet and the Caffe ML library. The main Caffe source tree includes LSTM, which we need, but does not include a CTC (Graves et. al., 2006) loss function implementation. For that, we can rely on a fork of Caffe which does include a realization of CTC. The CTC in this fork implements the Warp-CTC method created by Baidu (Amodei et. al., 2015).

With SparkNet and Caffe with Warp-CTC installed on the analytic pool, we can define a neural net pipeline for that pool. The specific pipeline we will define is similar to the pipeline defined in the study by Amodei et. al. (2015) to transcribe audio. Transforming audio to text using ML not only needs to be able to have previously recognized audio affect the expectation of audio heard later; audio actually heard at a given moment can alter aspects of previously recognized audio. To do either, an LSTM is necessary, but to do both is beyond the capability of a unidirectional LSTM and a bidirectional LSTM is required.

There are three layers in the neural network implemented over SparkNet. These are as follows:

- convolutional neural network
- bidirectional multimodal long short-term memory
- connectionist temporal classifier

The first layer is a convolutional neural net (CNN) (LeCun, Kavukcuoglu, & Farabet, 2010) which takes a first pass at extracting features. It is attempting to recognize characters of text by recognizing the invariant features characteristic of letters in the words. In so doing, it learns what otherwise would be hand-coded optimizations for an imaging system.

The second layer is a multimodal Bidirectional Long Short Term Memory (BLSTM) (Graves, Fernández, & Schmidhuber, 2006) that improves on that recognition by identifying likely words. The LSTM enables arbitrarily distant features within the area to impact on one another, so that context for any one feature need not be immediately adjacent to be recognized. The bidirectional aspect means that features seen later can serve as context for earlier-seen features as well as vice versa. Here, this means that newly recognized words can provide guidance to resolve ambiguities in words previously seen.

The third layer handling the image regions is a connectionist temporal classifier (CTC) (Graves et. al., 2006). A CTC is trained to recognize interrupted sequences, such as words which may have been broken into segments (extraneous spaces introduced by the scan, or hyphenization across a line break).

To train then only requires selecting a labeled subset of the data in the storage pool, subdividing that subset into training and testing data and issuing the command to train a model to the pipeline. Cassandra will manage any replication necessary to make that data available from the storage pool to the analytic pool.

5 Workflow

We envision the incremental accumulation of a corpus to be a distributed, parallel operation among multiple participants where there is no common calibration between the various cameras they are using as instruments. At any given time, one or more researchers may image a record from the collection. This image is submitted to a DRAS-TIC digital repository, assigned a UUID and a timestamp, and is stored as an object in Cassandra.

Capturing the image can be coordinated through HTML5 and a capable browser. A variation of the input tag can capture image data, and it will wait for the camera to take an image in the same way that more familiar uses of the input tag in forms wait until the tab or enter key is pressed to capture text.

DRAS-TIC invokes the extraction neural network pipeline through Brown Dog. Brown Dog sees the neural net layers in the insertion process as encapsulated by Python and packaged in a Clowder container which organizes their invocation. DRAS-TIC can trigger the

Clowder container to commence processing at intervals; when it does, a group of ingested records will be put through the insertion process. This runs them through the neural net pipeline implemented using Caffe with Warp-CTC across SparkNet. Each of these layers is progressively refining the view of the text within the ingested image.

The multiple outputs of the pipeline are a text representation of the image annotated with POS tags, chunked, and linked by relationship markers, all described as document interpretation acts. Brown Dog returns these to DRAS-TIC, which can store these outputs in its PostgreSQL database. The document interpretation acts organizing what is stored there by DRAS-TIC capture the classifications generated by the neural network along with manual or algorithmic corrections or amendments to them.

We envision the user being able to submit images as a batch process, meaning that in normal operation users will not wait for the system to finish processing one image before they capture the next.

6 Discussion

There are concerns with the training data, in that the images all went through QA as part of a digitization effort and will not be noisy enough to train for suboptimal images. Algorithmically generated noise – speckles, shadows, and skews – will need to be added to the data to simulate actual use conditions. The testing data did not all pass QA, which makes it more useful for encountering actual suboptimal images, but by the same token was generated as part of a professional digitization effort, so noise will need to be added to the testing data as well.

In addition, the algorithmically generated labels for the training data derive ultimately from the work of the community building the Penn Treebank, and whatever biases are introduced by that community are currently not accounted for. This is an issue which must be addressed, as there is no assumption about a neutral point of view covering the corpus as a whole and hence the imposition of a point of view from a community outside of the one centering on our collection should be noted and possibly corrected for.

It is reasonable to expect parallel language constructs between the training data and test and actual data, due to the definition of the sets. There are 118 volumes of AFL-CIO newspapers with text, covering the same set of years as the actual data and produced by divisions of the same organization, and read by many of the individuals who had responsibility for producing the records in the actual data. The evolution of the corpus as the neural network training data labels better reflect the language model of the community will provide data on differences between this corpus and one built from text judged as "objective" by an editorial community such as the editors of the *Wall Street Journal*.

7 Conclusion

We have found support for the possibility of a system which can incrementally build an annotated corpus by learning, recognizing, and describing text representations, annotations, and relationships between words in images. There are no requirements for proprietary software solutions, and we can bootstrap the process by algorithmically transforming existing labeled data to create the training data for its neural layers. It is deployable as a set of Clowder containers on a Brown Dog server cluster and can integrate with the storage pool of a collection managed by DRAS-TIC. SparkNet simplifies both dataflows and deployment, as the Clowder containers can leave in-memory management of models and data to Spark and management of storage likewise can be the responsibility of Cassandra.

We can see here an example of how Computational Archival Science can help theorize new relationships between communities and the memory institutions that serve them by collaboratively constructing a corpus based on their collection using entirely open source and can be replicated by other institutions. It will be interesting to see how this system as it is prototyped and developed affects the perceptions of the institution and community vis-a-vis each other.

References

Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., … Zhu, Z. (2015). Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *ArXiv E-Prints*, arXiv:1512.02595

Bohannon, A., Pierce, B. C., & Vaughan, J. A. (2006, June). Relational Lenses: A Language for Updatable Views. In *Proceedings of the Twenty-Fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (pp. 338–347). New York, USA: ACM.

Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4, 357–370.

Clarkson. K. L. (1984). *Algorithms for Closest-Point Problems (Computational Geometry)*. Ph.D. Dissertation. Stanford University, Palo Alto, CA.

Craven, M., & Kumlien, J. (1999, August). Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology* (pp. 77–86). Cambridge, MA : AAAI Press.

Dhingra, B., Jin, Q., Yang, Z., Cohen, W. W., & Salakhutdinov, R. (2018). Neural Models for Reasoning over Multiple Mentions using Coreference. *ArXiv E-Prints*. Retrieved from https://arxiv.org/abs/1804.05922

Gerber, D., Hellmann, S., Bühmann, L., Soru, T., Usbeck, R., & Ngomo, A.-C. N. (2013, October). Real-time RDF extraction from unstructured data streams. In *International semantic web conference* (pp. 135–150). Berlin, Heidelberg: Springer.

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on machine learning* (pp. 369–376). New York, USA: ACM.

Jansen, G. & Marciano, R. (2016, December). *DRAS-TIC Measures: Digital Repository at Scale that Invites Computation (To Improve Collections)*. Coalition for Networked Information, Washington, DC.

Jansen, G., Padthi, S., Marciano, R., McHenry, K. (2016, October). Designing Scalable Cyberinfrastructure for Metadata Extraction in Billion-Record Archives. In *iPres 2016 13th International Conference on Digital Preservation* (pp. 117-185), Bern.

Hall, J. D. (2005). The long civil rights movement and the political uses of the past. *The Journal of American History*, *91*(4), 1233–1263.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

IMLS. (2017). *Improving Fedora to Work with Web-Scale Storage and Services*, proposal abstract. Retrieved from https://www.imls.gov/sites/default/files/grants/lg-71-17-0159-17/proposals/lg-71-17-0159-17-full-proposal-documents.pdf

Isaac, L., & Christiansen, L. (2002). How the civil rights movement revitalized labor militancy. *American Sociological Review*, *67*(5), 722–746. Retrieved from http://www.jstor.org/stable/3088915

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675-678). New York, USA:ACM.

Kissos, I., & Dershowitz, N. (2016, April). OCR error correction using character correction and feature-based word classification. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)* (pp. 198-203). New York, USA: IEEE.

Kramer, J., & Magee, J. (2007, May). Self-managed systems: An architectural challenge. In *2007 Future of Software Engineering* (pp. 259–268). Washington, DC, USA: IEEE Computer Society.

Lakshman, A., & Malik, P. (2010). Cassandra- A decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, *44*(2), 35–40.

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (pp. 253–256). New York, USA: IEEE.

Loper, E., & Bird, S. (2002). NLTK: the natural language toolkit. *arXiv preprint cs/0205028*.

Ludäscher, B., Marciano, R., & Moore, R. (2001). Preservation of digital data with self-validating, self-instantiating knowledge-based archives. *SIGMOD Record, 30*(3), 54–63.

Marciano, R., Lemieux, V., Hedges,M.,Esteva, M., Underwood, W., Kurtz, M. & Conrad, M.(2018). Archival records and training in the age of big data. In P. Johnna, S. Lindsay., J. Paul, & B. John,(Eds.),*Re-Envisioning the MLS: Perspectives on the future of library and information science education* (pp. 179-199). Bingley: Emerald Publishing Limited.

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank (No. MS-CIS-93-87). *Computational Linguistics, 19*(2), 313–330.

Marini, L., Gutierrez-Polo, I., Kooper, R., Satheesan, S. P., Burnette, M., Lee, J, … McHenry, K. (2018, July). Clowder: Open Source Data Management for Long Tail Data. In *Proceedings of the Practice and Experience on Advanced Research Computing (PEARC '18).)*. New York, NY, USA: ACM.

Martín, A., Ashish, A., Paul, B., Eugene, B., Zhifeng, C., Craig, C., … & Matthieu, D. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Mattingly, I. G. (1972). Reading, the linguistic process, and linguistic awareness. In J. F. Kavanagh & I. G. Mattingly (Eds.), *Language by ear and by eye: The relationship between speech and reading.* Oxford, England: Massachusetts Inst. of Technology P. McHenry, K., Bradley, S., Dietze, M., Kumar, P., Lee, J., Marciano, R., Marini, L., … Sullivan, B. (2017). DIBBs Brown Dog - The Need for and Challenges of a Science Driven Data Transformation Service. In *DIBBs PI Workshop*. Retrieved from https://dibbs17.org/report/Papers/1261582paper.PDF

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., … Talwalkar, A. (2016). Mllib: Machine learning in apache spark. *Journal of Machine Learning Research*, *17*(1), 1235–1241.

Miwa, M., & Bansal, M. (2016). End-to-end relation extraction using LSTMs on sequences and tree structures. *CoRR*, *abs/1601.00770*. Retrieved from http://arxiv.org/abs/1601.00770

Moritz, P., Nishihara, R., Stoica, I., & Jordan, M. I. (2015). SparkNet: Training deep networks in spark. *ArXiv E-Prints*. Retrieved from https://arxiv.org/abs/1511.06051v1

Neuberg, B. (2017). Creating a Modern OCR Pipeline Using Computer Vision and Deep Learning. Retrieved from https://blogs.dropbox.com/tech/2017/04/creating-a-modern-ocr-pipeline-using-computer-vision-and-deep-learning/

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011, June). Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (pp. 689–696). USA: Omnipress.

Padhy, S., Jansen, G., Alameda, J., Black, E., Diesendruck, L., Dietze, M., … Marciano, R. (2015, October). Brown Dog: Leveraging everything towards autocuration. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 493-500). New York, USA: IEEE.

Padilla, T., Allen, L., Varner, S., Potvin, S. Roke, E. R., Frost, H. (2018). *The Santa Barbara Statement on Collections as*

Data, Always Already Computational. Retrieved from https://collectionsasdata.github.io/statement/

Pasin, M., & Bradley, J. (2015). Factoid-based prosopography and computer ontologies: Towards an integrated approach. *Literary and Linguistic Computing*, *30*(1), 86–97.

Paul, D. B., & Baker, J. M. (1992, February). The Design for The Wall Street Journal-based CSR Corpus. In *Proceedings of the Workshop on Speech and Natural Language* (pp. 357–362). Stroudsburg, PA, USA: Association for Computational Linguistics.

Ren, J., Hu, Y., Tai, Y.-W., Wang, C., Xu, L., Sun, W., & Yan, Q. (2016, March). Look, Listen and Learn–a Multimodal LSTM for Speaker Identification. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Ristoski, P., & Paulheim, H. (2016, October). RDF2Vec: RDF graph embeddings for data mining. In P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch, F. Lecue, & Y. Gil (Eds.), *The Semantic Web – ISWC 2016* (pp. 498–514). Cham: Springer International Publishing.

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

Satheesan, S. P., Alameda, J., Bradley, S., Dietze, M., Galewsky, B., Jansen, G., … McHenry, K. (2018, July). Brown dog: Making the digital world a better place, a few files at a time. In *Proceedings of the Practice and Experience on Advanced Research Computing* (p.1–8). New York, USA: ACM.

Schwartz, D. L. (1999). The productive agency that drives collaborative learning. In P. Dillenbourg (Ed.), *Collaborative learning: Cognitive and Computational Approaches. Advances in Learning and Instruction Series* (pp. 197-218). New York: Elsevier.

Shi, B., Bai, X., & Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,39*(11), 2298–2304.

University of Maryland. (2016). *The George Meany Memorial AFL-CIO Archives at The University Of Maryland: Guide To Collections*. Retrieved from http://lib.guides.umd.edu/c.php?g=327462&p=2196193

W3C. (2015). *Linked Data Platform 1.0,* W3C recommendation. Retrieved from https://www.w3.org/TR/ldp/

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., … Stoica, I. (2016). Apache Spark: A unified engine for big data processing. *Communications of the ACM*, *59*(11), 56–65.