Learning to Filter Documents for Information Extraction using Rapid Annotation

Carlos A. Aguirre, Sneha Gullapalli, Maria F. De La Torre, Alice Lam, Joshua Levi Weese, William H. Hsu

Laboratory for Knowledge Discovery in Databases, http://www.kddresearch.org
Department of Computer Science
Kansas State University, Manhattan, KS 66506
{caguirre97|snehag|marifer2097|lamalice16|weeser|bhsu}@ksu.edu

Abstract

Corpus-driven approaches to information extraction from documents face problems of relevance determination, namely determining which documents are of requisite type, structure, and content for a specified query and context. In this paper, we discuss the problem of *learning to filter* documents crawled from the web with respect to such relevance criteria, and in particular how to annotate document corpora for supervised classification learning approaches to this problem. For context, we describe a system aimed at extracting experimental data from scientific publications, with the long-term goal of extracting procedural information from relevant sections on experimental methodology. We consider motivating use cases for our learning filter, using the documents passed by the filter: marking up sections (or passages); capturing entities and relationships; and explaining to a domain expert why a document is relevant. These distinct use cases make the annotation task multifaceted. Our approach focuses on speeding up annotation in learning to filter while minimizing loss of precision or recall on the learning task, using a reconfigurable user interface. We develop such an interface, report on its use in tandem with classification on a real extraction task, and discuss extensions of this work to visual scene filtering and annotation.

1 Introduction

This paper describes an annotation-based approach to learning to filter documents crawled from the web for structured or semi structured information extraction (IE) tasks. These tasks themselves may involve further machine learning and lower-level markup of the documents or the text payload extracted from them. We review extant learning approaches and their implications for learning representation, feature construction and selection, and resource-bounded learning – particularly utility-driven active learning and cost-driven semi supervised learning when annotation is a dominant component of cost. We also consider the question of how to assess transfer across subdomains of a unified corpus that spans many topics. This presents an interesting problem of determining the range of salient topics given an extraction task, possibly via a constrained form of topic modeling.

The primary novel contribution of this work is the development of an annotator user interface that enables documents to be previewed for quick rejection judgements related to type (whether they are papers), structure (whether they are of requisite length, contain figures or other elements of interest, and are formatted into sections), and content (entities and relationships of relevance to the procedural IE target, and procedural information itself).

1.1 Information Extraction (IE) Task

This work focuses on structured relationship extraction and inference from free text, using supervised learning for classification to determine which tokens demarcate the start and end of sections, passages, and chunks. This methodology is now a standard approach in shallow parsing, also known as chunk parsing. In this section, we present the framing task and the central research problem of this paper: how to acquire the corpora to be marked up for training the needed classifiers for the extraction task. We discuss how this process can be bootstrapped by acquiring many documents and training more basic, document-level classifiers to filter documents as relevant ones to annotate further.

Goal: Extracting Procedural Information
A long-term goal of this research that defines the IE task is to develop an autonomous reading capability by automating the extraction of recipes for materials of interest. These *recipes* consist of raw ingredients and their proportions and quantities, manufacturing plans, and information about timing.

The specific application domain we are investigating involves nanomaterials manufacturing processes based on defined reactants, products, and steps. These will be based on an ontology partly developed through knowledge elicitation from domain experts and partly extracted from corpora.

The technical objectives include:

1. Extracting materials, chemicals (reactants, products, and waste products), primitive process steps, and specified functions by name from text.
2. Extracting associated figures and other images from sections of a published paper or other article, and relating these images to a representation of a recipe for a nanomaterial.
3. Creating one or more recipes for each paper to interrelate inputs, products, and processes
4. Given an expert-provided process for creating a specific nanomaterial, matching the extracted recipes to identify deviations from the specified baseline process. This will identify deficiencies in the extraction technique, enabling improvements including the automated discovery of unique component terms once the extraction method is refined and validated.
5. Given the newly-extracted process model, and a database of recipes, identifying the common and uncommon elements in the new model
6. Given quantitative and qualitative similarity measures, and expert-provided use cases, developing an analogy-based system for transferring understanding (in the form of recipe matching, prototypes, etc.) from one nanomaterial to another. An example of this is adapting a process for a copper material to a tungsten material.

These steps require us to formalize the concept of a recipe as a precursor to a plan, containing the identifiable preconditions, material inputs, products, steps, and a representation of their interrelationships.

Prerequisite: Acquiring Relevant Documents
The proposed approach supports gathering of data by means of a flexible web crawler that we have developed. It will require digestion of online sources of free text in basic publication formats (PDF and HTML), adaptation of existing open-source and other software for end-to-end natural language processing (NLP) operations, including entity matching and gazetteer-based named entity recognition as needed.

Research and development objectives may include any of the following as required to achieve the performance goals of the system: sense induction and disambiguation; passage, snippet, and section extraction; relevance determination (feature extraction/construction); and deduplication. These tasks are defined in a manner similar to their analogues in question answering (QA), knowledge base population (KBP), and wikification, extant research areas in the text analytics, information extraction, and information retrieval communities.
One of the primary objectives is to identify the relevant features and eliminate the irrelevant ones to filter the documents to improve the prediction accuracy. For this purpose, we use two approaches of selective learning called feature selection and example selection. The feature selection approaches result in only the features that gives useful information amongst all the features. We have used a list a gazetteer vocabulary which identifies the documents into various categories such as relevant, irrelevant papers (posters, advertisements etc.).
In example selection, rather than eliminating the information, we make use of informative instances in the training data. We have used the TFIDF vectorizer from the scikit-learn Python machine learning library (Pedregosa, et al., 2011) for the relevance task (evaluates the importance of a word in the document). Also, we applied the k-fold cross validation which divides the whole set of documents into k batches and then used the (k-1) folds for the prediction function learning and the other one for testing. This is being done on a large set of 30K documents in our test bed. It is observed that the speed up is quite high, almost doubled using the interface for annotating compared to the traditional annotation techniques. We try to optimize the concept of learning to filtering synchronizing the outputs in both the ways.

The key improvements sought in this research include using machine learning to filter, rank, and prioritize documents that are found. This driving problem provides a rationale for the annotation-driven framework presented in this paper, into our approach for learning to filter, and into the development of more efficient user interfaces for annotation.

1.2 Learning to Filter

Learning to filter is the task of training a classifier to determine whether a given document meets informal criteria such as the abovementioned ones of type, structure, and content. It is an analogue of *learning to rank* in information retrieval – the problem of using supervised, semi-supervised, or reinforcement learning to product an ordering of documents for some search task.

Given a downstream annotation problem of marking up text extracted from documents that include peer-reviewed articles, the task of learning to filter becomes whether a given document is one such article and whether it is on-topic for the extraction of markup training data.

1.3 Need for Fast Annotation

Training a learning filter may require many documents to be sifted through to obtain positive examples (properly-formatted, relevant, on-topic documents). When there is a significant degree of class imbalance – specifically, a low true positive rate – this annotation task can become very time-consuming. The amount of time spent examining each document is also a factor. We seek to reduce this by streamlining the annotation process.

2 Background and Related Work

2.1 Learning to Filter

The accuracy of learning algorithms decreases with the presence of irrelevant information, making the pre-processing of training data highly important. For a learning system with the purpose of document classification, the objective of focusing on the most relevant information has become increasingly

challenging since there is a large volume of unrelated information extracted from a crawl of the World Wide Web. Methods for handling datasets that contain large amounts of irrelevant information entail the selection of relevant features and relevant examples (Blum & Langley, 1997). In selective learning, feature selection is defined as selective attention and example selection is defined as selective utilization (Markovitch & Paul, 1993).

Feature selection is the process of optimally reducing the feature space by choosing a subset of features based on certain criteria, this process speeds up the learning algorithm, and improves the predictive accuracy of a document classification algorithm. Feature subset generation can be done through sequential forward or backward selection or by randomly generating a subset. A candidate subset can then be evaluated using validation data. (Liu & Motoda, 1998) Relevant example selection is another important component of selective learning; it reduces the effect of irrelevant information by selecting examples that aid the training process of the algorithm. With a large amount of training examples, relevant example selection improves computational efficiency and the rate of learning. Schemes for both feature and relevant example selection involve filter methods, wrapper methods and methods that are part of the learning algorithm (Raman & Ioerger, 2003).

To implement both feature selection and relevant example selection, in this research, we use a gazetteer containing terms relevant with the given topic to train the learning classifier. Term frequency-inverse document frequency (TF-IDF) was used as a term weighting scheme to improve the retrieval of information by determining the occurrence probabilities of terms. TF-IDF is a measure that multiplies the direct estimation of the frequency of a term when normalized by the total frequency in the document (TF) by the log of the inverse probability. (Aizawa, 2003) Given a document or a term, TF-IDF expresses the significance of the component as a product of the frequency and the amount of information that it represents. As an estimator of prediction error for the classifier, the k-fold cross validation (k-cv) was used. 30,963 pdf documents were divided into folds, the k-cv estimation of the error is the average value of the errors committed in each fold, and depends on the training data and the number of folds. These measures improve filtering in the pre-processing of the data and the classifier's predictive results for document classification.

2.2 Use Cases for Learning Filter

A learning filter returns documents that meet some specified predicate, but its ancillary output can be helpful in inference tasks such as downstream extraction tasks or explaining subsequent results to a user.

Section or Passage Extraction
One basic downstream task is to highlight the section or passage (snippet) where desired procedural information resides, such as an Experimental Methods section or a header less section of text containing a description thereof.

Named Entity Recognition and Relationship Extraction
A related step to marking up free text and segmenting passages of interest is the identification of named entities – in our case, names of compounds, waste products, etc. – that are related to a chemical engineering process. Other related tasks include extracting relationships among these entities and the temporal and numerical information mentioned in Section 1.

Explanations
Finally, it is often useful to domain experts for an IE system to explain the relevance of an extracted entity, relationship, or in this case procedures represented as tuples and sequences. A learning filter allows the criteria applied to each document be presented to the domain expert along with concrete evidence supporting the inclusion of the resulting documents.

3 Methodology

3.1 Speeding up Annotation

Usability Criteria
Our approach to the user interface was with usability as the highest priority. Working with diverse documents that were crawled from different sources presents a problem to the rendering of the preview of the pages in the documents due to the lack of uniformity in the format of the documents. For this reason, finding an existing free-available annotation tool that would fit our purpose was not possible. The analytic method best described by (Burghardt, 2014) was used in the development of the annotator. This was the ideal choice with regards of taking into account the target user, since the users are ourselves. We designed the user interface to satisfy the need of a fast and responsive tool to annotate a great amount of training data while keeping it easy to configure for different future projects.

While designing the annotator, an important feature for the final product was the usability degrees of freedom. The tags in the annotator interface can easily be customized to meet the requirements of the specific project. Dynamically adding tags for unexpected types of documents can increase the functionality of the annotator. While having the tags available as clickable buttons is the current scheme, the option of using keyboard shortcuts is available to the user which would potentially add to the speed of the annotator and the usability of the user interface. The targeted data type can be modified to accept images or scenes and have snippets for faster sorting. Individual snippets can offer a closer look to important passages of the document. The goal is to single out relevant passages of the document. There can be more than one snippet in a document depending on the relevance with respect to the next idea. (Zhou, Yu, Smalheiser, Torvik, & Hong, 2007) These snippets would provide the user with an overview of the object to annotate for an easier and faster understanding of the object at hand. While queuing documents is a main feature of the annotator, pre-queuing future documents for easy handling of the program would increase its efficiency in real time. And finally, it allows for TF-IDF specific results and

gazetteer highlighting. We use TF-IDF to determine the top words of the document and the gazetteer to represent the words that are considered relevant. The implementation of these features aids the user to annotate faster while minimizing inaccurate annotations.

3.2 Objectives for Learning

The purpose of machine learning for this task is to facilitate greater automation of the process of acquiring documents, filtering them based on past experience to quickly reject:

- invalid documents: corrupt or illegible files
- non-articles: those that are not the peer-reviewed paper itself (e.g., posters based on a paper, publisher forms, advertisements)
- irrelevant articles: those that are off-topic

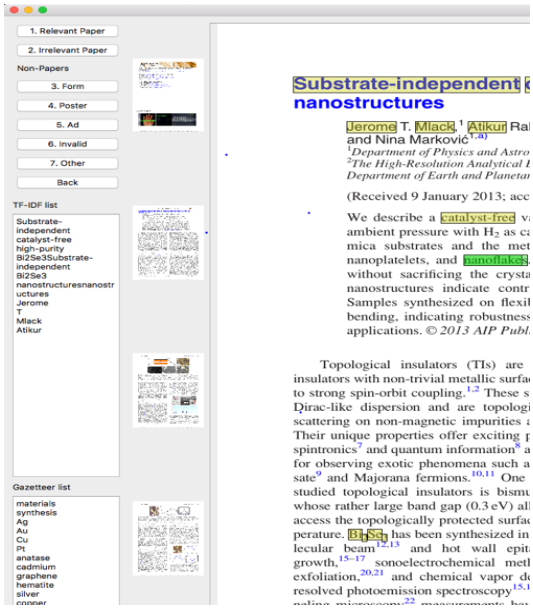

Figure 1. Graphical user interface (GUI) for annotator. Green highlight is gazetteer, yellow represents TF-IDF

We treat this problem as a prerequisite task to document analysis and markup. Figure 1 depicts a graphical user interface for collecting this information by previewing documents. The annotator buttons record class labels for each document, which is previewed in the GUI.

Our primary objective for learning is to be able to train a classifier based on **both textual and non-textual features**. *Textual features* include keywords in the document from a gazetteer, or unigrams and bigrams of common words encountered in the domain literature. *Non-textual features* include visual ones, such as the aspect ratio of a document, the ratio of payload to tag metadata and other markup, the frequency of figures and other embedded data, keywords in the document from a gazetteer, or unigrams and bigrams of common words encountered in the domain literature. The results of reasoning over textual features given a domain ontology (i.e., ontology-aware inference) may also constitute non-textual features.

3.3 Crawler

For generating our document corpus, a custom web crawler was created with a multi-stage filtering system, as well as a general acceptability for niceness. Currently, the crawler only contains filtering for URLs. This includes ad and spam detection, as well as media type (pdf, doc, etc.). Filtering the content of URLs is offloaded to a separate script to improve crawler efficiency. The niceness property of the crawler is enforced through restrictive scheduling of scraped URLs. As such, robots.txt policies are obeyed. Throttling the crawl based on the URL is done on both the page and domain level by using priority queues and a minimum time frame between attempts. This prevents the crawler from attempting to fetch a page or a page from a domain too frequently.

The crawl run for the corpus in this paper began using the 13 seeds below. These seeds are mostly paper or abstract links of relevant publications. From these starting seeds, we extracted 30,963 PDF files. These documents were then filtered for relevance by checking for the presence of gazetteer items (TF-IDF), leaving 19,419 documents. This pre-filtering stage is only intended to exclude documents that have no vocabulary relevant to our topic and is not intended to exclude any documents that are not full papers (abstracts, presentations, posters, etc.).

1. https://www.researchgate.net/publication/247949042_Large-Scale_Synthesis_of_Uniform_Silver_Nanowires_Through_a_Soft_Self-Seeding_Polyol_Process
2. http://pubs.acs.org/doi/abs/10.1021/nl048912c
3. http://pubs.acs.org/doi/abs/10.1021/nn400414h
4. http://pubs.acs.org/doi/full/10.1021/acs.nanolett.5b02582
5. http://onlinelibrary.wiley.com/doi/10.1002/cjoc.201400518/abstract
6. http://pubs.acs.org/doi/abs/10.1021/cr100275d
7. http://onlinelibrary.wiley.com/doi/10.1002/anie.201100087/abstract
8. http://pubs.acs.org/doi/abs/10.1021/acs.jpclett.5b02123
9. https://www.researchgate.net/publication/230739689_Defining_Rules_for_the_Shape_Evolution_of_Gold_Nanoparticles

10. http://pubs.acs.org/doi/abs/10.1021/ac0702084
11. http://pubs.rsc.org/en/Content/ArticleLanding/2012/RA/c2ra21224b#!divAbstract
12. http://www.mdpi.com/1996-1944/3/9/4626
13. http://pubs.acs.org/doi/abs/10.1021/la050220w

4 Experiment Design

Two annotation processes were tested. The first entailed simply previewing PDF files in Adobe Acrobat Reader and sifting them manually into "relevant" and "irrelevant" directories. The second used our fast annotator GUI, which previews documents but provides a panel containing a button for each label, and cues up the next PDF file while the current one is being viewed. This interface is capable of writing results directly to training data in a database or file, but for fairness of comparison we measure the overhead of sifting the PDF file in Table 2 as well.

The annotator experiment used 3 annotators each of whom received a set of 2520 documents from among the 30K crawled documents: 1260 to label using the slow annotation process and 1260 to label using the fast annotator. Later these results are used for generating wide word vectorizers using tf-idf. A large bag of words is generated from a small batch of 105 files which resulted in 7633 words. These words are words are considered as features and as baseline to generate high dimensional tf-idf vectors for each document. These results are zipped with a class label determined by best of three (3) annotations. Later this data is analyzed using various algorithms like Logistic, Naïve Bayes, Random Forests and results are presented below

5 Results

Tables 1 and 2 show the results (accuracy, weighted average precision, average recall, F1 score, and area under the ROC curve) for the two variations of the annotator, using 10-fold cross validation. **Bold face** indicates the better of the two sets of results.

The inducers compared are:

- Logistic: Logistic Regression
- IB1: Nearest Neighbor
- NB: Discrete Naïve Bayes
- RF: Random Forests

Table 1. Results for slow (manual) annotator.

Inducer	Acc	Prec	Rec	F1	AUC
Logistic	**75.2%**	0.711	**0.752**	0.709	0.640
J48	**78.3%**	0.782	**0.784**	**0.783**	**0.688**
IB1	79.9%	0.788	0.799	0.792	0.712
NB	**74.2%**	**0.790**	**0.742**	**0.757**	0.759
RF	79.4%	0.801	0.795	0.736	0.841

Table 2. Results for fast annotator.

Inducer	Acc	Prec	Rec	F1	AUC
Logistic	69.3%	**0.764**	0.693	**0.719**	**0.664**
J48	77.9%	**0.785**	0.779	0.782	0.668
IB1	**83.8%**	**0.824**	**0.838**	**0.827**	0.695
NB	71.7%	0.789	0.718	0.742	**0.785**
RF	**83.3%**	**0.825**	**0.833**	**0.788**	**0.862**

The average time required for fast annotation was 5,246.8 seconds vs. 18,413.4 seconds for slow annotation, with insignificant differences in precision or AUC, slightly lower accuracy, and lower recall. The fast annotator has a significant speed up of 251%.

6 Conclusions and Future Work

6.1 Further Speedup

The results indicate potential gains to be realized by streamlining the annotator interface, but the minor speedup attained so far achieves only a few of these gains. Figure 2 shows how the current document previewer is similar to Adobe Acrobat Reader, in that it renders a single page at a time to a full-scale viewer, and includes small thumbnails of other pages. One possible use of machine learning in continuing work is to select features that are relevant or potentially relevant (through active learning) to present "at a glance" summary information, or more comprehensive, holistic previews, to the user.

Figure 2. Annotator interface showing PDF document preview.

6.2 Active, Transfer, Semi supervised Learning

A central focus of continuing work is the further development of this test bed for linguistic annotation-based machine learning, to incorporate aspects of active learning, transfer learning, and semi-supervised learning.

Active learning has the potential to benefit the efficiency of supervision, such as in query-by-example protocol where the documents presented to an annotator through the interface are selected based on incremental learning results to date. As mentioned above, however, it can also help select relevant elements of interest (keywords, figures, pages, snippets) to preview. This presents an interesting domain for application of utility-theoretic active learning in computer vision.

Another challenge we face is that seed documents are relatively scarce for the domain of application, so that transfer learning by transduction may be of use. Furthermore, the clear majority of documents acquired will remain unlabeled even if the annotator is sped up by a significant factor, so that semi supervised learning (with class imbalance) is a framing problem.

A final area that is supported by our current research results and consists of some open research problems is that of identifying anomalous aspects of inferred processes relative to those specified by domain experts. In continuing work, we will investigate transferability of processes using traditional intelligent systems approaches as derivational and transformation analogy with newer tools such as transfer learning, with the end goal of synthesizing them into a modern analogy-driven learning and reasoning component.

Acknowledgments

This work was funded by the Laboratory Directed Research and Development (LDRD) program at Lawrence Livermore National Laboratory (16-ERD-019). Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344. This is also supported by the U.S. National Science Foundation (NSF) under grants CNS-MRI-1429316 and EHR-DUE-WIDER-1347821.

We thank Margaret Rys of the Department of Industrial and Manufacturing Systems Engineering (IMSE) at Kansas State University for an independent usability critique and helpful advice on ergonomic design.

References

Aizawa, A. (2003). An information-theoretic perspective of TF-IDF measures. *Information Processing & Management, 39.1*, 45-65.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *97.1*, 245-271.

Burghardt, M. (2014). *Engineering Annotation Usability Toward Usability Patterns for Linguistic Annotation Tools (Ph.D. dissertation).* Regensburg, Germany: Universität Regensburg.

Liu, H. & Motoda, H. (1998). Feature extraction, construction and selection: A data mining perspective.

Markovitch, S., & Paul, S. D. (1993). Information filtering: Selection mechanisms in learning systems. *10.2*, 113-151.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830. Retrieved from http://scikit-learn.org

Raman, B., & Ioerger, T. R. (2003). *Enhancing Learning using Feature and Example selection.* College Station, TX, USA: Texas A&M University.

Zhou, W., Yu, C., Smalheiser, N., Torvik, V., & Hong, J. (2007). Knowledge-intensive Conceptual Retrieval and Passage Extraction of Biomedical Literature. *Proceedings of the 30th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR, 2007)* (pp. 655-662). Amsterdam, The Netherlands: ACM Press.