Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Creating and Visualizing Fuzzy Document Classification

Judith Gelernter Dong Cao Raymond Lu Eugene Fink Jaime G. Carbonell

School of Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
U.S.A.
gelern@cs.cmu.edu; caodshen@cs.cmu.edu; raylu@cmu.edu; e.fink@cs.cmu.edu; jgc@cs.cmu.edu

***Abstract*—Fuzzy classification ranks items by degree rather than assigning them either within or without of a category. The novelty of our work is in integrating fuzzy classification algorithms with an interface to visualize fuzzy results. An advantage of our algorithms' 'fuzziness' is that it provides additional information per retrieved result that helps in deciding whether to drill down to the document or skip it. An advantage of our interface is that it allows users to visualize those differences quickly. We have created a prototype that allows the retrieval of journal articles by content word or by ontology-supported browse categories that can be selected independently or in tandem. Journal articles in our digital library pertain to paleontology, but techniques demonstrated viable in indexing and ranking paleo-journal literature should apply to other knowledge domains with little modification.**

***Keywords*—fuzzy retrieval systems, fuzzy match, relevance, retrieval model, classification; information visualization, graphical user interfaces**

I. Background in Text Classification

Fuzziness has been used in statistics and in computer science to indicate the probability that something is true. Fuzziness is about uncertainty. The concept has been used in information retrieval to account for the data itself [1], for an ontology to aid in matching [2], for method of matching [3], for result visualization [4].

What do we really mean by fuzziness? In interviewing a selection of professionals who "self-identified as having aspects of uncertainty in their [daily] work", the definition of the concept most often mentioned was that of level of uncertainly [5]. The idea comes from fuzzy set theory, which was introduced in the 1960s by mathematician Zadeh [6]. Skeels et al. have identified five kinds of fuzziness: approximation, prediction, disagreement, incompleteness and credibility about the data source [5].

We invoke fuzziness in acknowledgement of fuzzy intuition on the part of the system user. Our user has a vague idea of what he wants, but generally will not know for sure until he sees it [7]. Our system will help by suggesting those items that might be the best match to what he wants, good matches and adequate matches. Our specific task is to classify each item such that it may belong to multiple categories partially rather than to one category absolutely. By "fuzziness" we mean the degree of relevance of an item to its classification and how it is displayed.

How crisp or fuzzy is the classification fit? The question has been examined over the years with respect to the importance of a term in a document, an early example being Salton, Wu, Yu [8]. A measure of the classification fit has been accomplished through document weighting such that some rules "outweigh" others in terms of strength of prediction of whether an item should belong to a category. For example, higher weighting might be assigned a term if it is found in a document repeatedly, or found in the title or abstract field, for example, than if it is found just once. Other models are referenced in Cimiano [9]. We discuss this in more detail below, and particularly in the "Classification crispness or fuzziness" section.

For automated text classification itself, we rely on standard methods and introduce a new measure of classification relevancy that suggests how good the classification fit seems to be. We introduce a method of weighting based on term frequency and location in the document. The documents are measured and ranked with respect to an absolute scale rather than with respect to each other. Our contribution is the High, Medium and Low paleo-document ranking within classification categories, and the visualization of such ranking.

Our methods for automated text classification follow the standard procedure: Sample data that are representative of the population are assembled, and then separated into a training set to create algorithms and test set to test algorithms. The

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

purpose of the classifier algorithm is to note a pattern common to some known group, and associate the item that manifests that pattern with its group. Patterns have been noted as heuristics, and then coded into a classifier algorithm in a method known as knowledge engineering. Alternatively, patterns may be determined from a large body of pre-classified items in a method known as machine learning. Some have found the accuracy from the application of machine learning to be comparable to knowledge engineering techniques [10], [11].

Machine learning or knowledge engineering: how should one choose? In machine learning, the classification effort goes into the construction of the learning algorithms that is prerequisite to the construction of the classifier and into labeling instances for training, rather than into making the classifier itself. In other words, the effort is put into how to extract rules from the training set, rather than whether these are the best rules and how to assemble them into a classification algorithm. Categories of classifiers include rule-based, probability-based, decision tree-based, multivariate regression-based, neural network-based, and nearest neighbor-based [12]. In the knowledge engineering approach, rules are engineered through manual inspection of the data, successive trials and refinements. The knowledge engineering approach, while it is weakened by its inflexibility (rules may require changing if categories are updated) and lack of portability (rules may need to be re-worked for each knowledge domain), may lead to better classifications. It also requires many fewer items in the training set to extract patterns. So it was more expedient for us to use the knowledge engineering approach for this study, though the fuzzy categorization would apply as well to machine learning.

Our fuzzy search tool is slated to be a beta-test site within the international e-science project, the GEON portal for the geosciences.[1] GEON includes a rich set of paleo-resources called the "PaleoIntegration Project". We were lucky enough to be asked by the GEON director to make our scholarly-literature related tool a model for all of the geosciences. As it is, some of our development choices were made for the sake of compatibility with other resources in GEON. Even our decision to create a workable interface without a good deal of preliminary human factors testing was driven by our desire to produce a prototype for GEON. We look forward to an online venue that promises a fair amount of user traffic, and from user evaluations we intend to improve the interface later.

II. PaleoSearch Architecture

Our system is diagrammed below. Fig. 1 shows schematically how a collection of journal articles (in .pdf) is stored separately from the parsed metadata (in .xml) that have been extracted for indexing. The classifiers use the metadata and refer to the ontologies to assign each article to organism name, time and region categories. Users may enter keyword(s) or select from browse menus for organism, time and region. Categories that appear in the menus are known yield results from the pre-classified corpus. Interface results display consists of article title with publication date, which is hyperlinked to a .pdf version of the article full text.

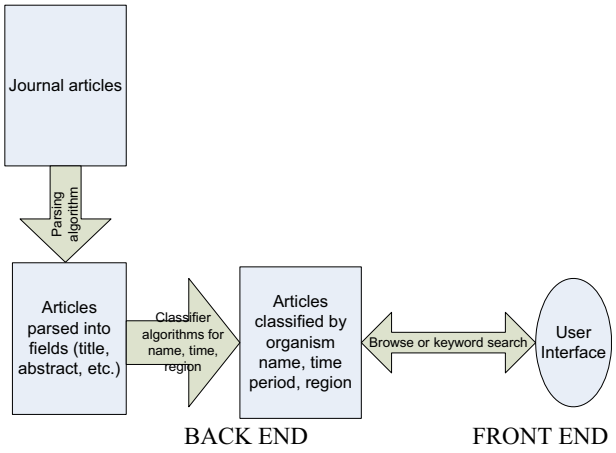

Figure 1 PaleoSearch architecture. The .pdf journal articles in the corpus are parsed into title, abstract, caption and other fields in .xml, and then are classified with the help of ontologies into organism name, time period and region categories. The articles are then retrievable by way of these categories.

The architecture is optimized for speedy return of results. Classification is time-consuming, particularly classification by organism name owing to the vastness of the bio-taxonomies. The trilobite ontology alone takes several hours to match against our 150 article training data sample. Rather than farm our matching to a supercomputer or re-structure searches to run in parallel, we simplified our indexing in two ways: (1) each item is classified upon entry to the database rather than when a query is input, and (2) the upper levels of a hierarchy as well as a more specific classification are saved along with the item to speed matching at the time of query entry. So for example, an article is classified as Archaeopteryx, but saved for possible matching with a keyword as Aves—Archaeopterygiformes—Archaeopterygidae—Archaeopteryx (that is, Class—Order—Family—Genus).

III. Classification for Information Retrieval

A. Triple indexes to search paleontology articles

Features for the prototype that would be useful to the discipline were discussed with paleontologist Christopher Noto, presently Visiting Professor of Grand Valley State University, Michigan. He expressed the need for articles within wide-reaching disciplines for the corpus. He also defined the three main points of access to be plants or animals, time period and region. The nature of the indices is essential: We use paleo-organism and paleo-times, but modern regions.

[1] http://www.geongrid.org

Why do we index region with place names that are modern? Paleo-regions such as Gondwana and Laurasia are so vast as to give paleontologists little feel for location. In addition, the literature that discusses organisms (that comprises our data sample) refers to modern regions where the fossils have been discovered or are housed. Algorithms exist that translate modern coordinates into paleo for a given period of time. But if we were to retrieve all articles that refer to modern regions in Europe and Asia under Laurasia, for example, the retrieval set of articles would be so vast that it would be unhelpful. These are the most compelling reasons why we use a modern index for region.

B. Information retrieval and the ontologies

Ontologies and taxonomies are types of controlled vocabulary. Ontologies have been found to improve information retrieval [13], especially for domain-focused document collections It has been known for decades that matching exhaustivity and specificity to the data set tends to improve retrieval [14]. Our preliminary tests with the ontologies showed us how to adjust the number of words in the ontologies (esp. for organism name and for region, as described later in the paper). Another fundamental question regarding any ontology is whether it should comprise not only words, but phrases as well. Ontologies for this study incorporate both. A species name, for example, is sometimes seen in a phrase lead by the genus.

Ontologies are limited by the words that constitute them. That is, different words per ontology mean that the choice of ontology will alter the system's retrieval properties. Ontologies with topic hierarchies have been derived from the data themselves [15]. Our research uses ready-made ontologies that are domain specific.

The ontologies in PaleoSearch were up-to-date at the time of absorption into the prototype. Whereas updating ontologies is, in most cases, an unavoidable requirement, the ontologies we use for organism name, time period and region are unlikely to change within the several year span of system use. Updating in this particular application, therefore, seems non-essential.

For the sake of compatibility with GEON, we adopted GEON ontologies for time and region. GEON has no organism ontology, however. PaleoSearch requires an organism ontology with species that are extinct as well as extant. We found a chronologically rich taxonomy in the Thomson Reuters Index to Organism Names (ION), and were given a sample of trilobites, but its plant taxonomy will not be available until at least 2010. So we have supplemented our ION Trilobite sample with animal and plant taxonomies from the Paleobiology database.

C. The sample: article selection and preparation for indexing

We selected a number of plant and animal paleo- taxa randomly.[2] Our choices are at different taxonomic levels, and there is some overlap (Allosaurus and Apatosaurus are within the order Saurischia, for example). We used these pre-selected taxa as keywords to collect 200 articles from open source and proprietary databases to which had access. Articles come from databases including Academic Search Premier, JSTOR, BioOne, Web of Science, PubMed, AnthroSource, Proceedings of the National Academy of Science (PNAS), PLoS One, and Google Scholar. The articles are taken from journals in fields of geology, ecology, biology (including evolution, anthropology, anatomy, zoology and botany), and chemistry, as well as paleontology.

We stratified the collected articles according to organism to make a 150-article training set and a 50-article test set. The by-product of mixing articles among journals and disciplines is that it produces a potentially more generalizable sample. The different disciplines use some of their own vocabulary to exercise our ontologies. The different layout formats of various journals challenge our parse algorithm.

1) Article preparation. To change the default, adjust the template as follows. The articles were downloaded in .pdf. Most were exported manually to .xml using Adobe Acrobat. We decided based on manual inspection that we needed to extract the metadata for classification from certain fields. The fields we defined are: title, abstract, year, university (that is, where authors come from), captions, body text, footnotes, keywords, and references.[3] Very few of the articles had author-supplied keywords, but for those that do, keywords are quite useful for classification.

We had hoped to use .xml tag divisions to help parse each article into fields. However, only the few articles which were downloaded in native .xml form had tags that were genuinely useful. The tags introduced by Adobe from the conversion between .pdf and .xml, while they subdivide the articles, do not distinguish but only separate one area of text from another. So to distinguish title from abstract from references, and so on, we inspected the articles manually and created heuristics to code.

2) Our parse algorithm. Our heuristics for the title field range from simple, to requiring that another program be invoked. Simple heuristics use labeling from within the article. For example, an article abstract begins often with the word "Abstract," and the reference section often begins

[2] For plants: Ginkgo, Cycad, Lepidodendron, Sequoia, Conifer, Glossopteris; For animals: Eurypterid, Saurischia, Archaeopteryx, Allosaurus, Homo habilis, Anthropoidea, Trilobite, Archelon, Apatosaurus, Ammonite

[3] We refer to this later as the "Lu Parser" for its developer, Raymond Lu of Carnegie Mellon University.

"References". More complicated means are necessary to isolate the title. To isolate the title to make title field metadata, first, we submit each article to an open source program to convert it to .html.[4] This open source program introduces font information that is helpful in distinguishing the title. In cases where there is no distinguishing font information, we subject the article to our heuristics to help find or, at last resort, generate a title. For example, we will take for the title field the first phrase of seven or more non-numerical words that contain a colon, or, we will take for the title field the first phrase of seven or more non-numerical words that appear before the word "Abstract," or "Author(s)".

Irregularities in the metadata as minor as an extra space or a misplaced ">" throw off our algorithm and prevent an article from being run through the Lu parser. Some of the articles did not parse properly and had to be discarded. The total number of articles for the training set thus comes to 147.

D. The classifier algorithms

We coded three separate algorithms to classify paleo-related articles based on three sets of heuristics. One classifier is for organism name with the name taxonomy, another is for geologic time period with the GEON time ontology and a third is for region with the GEON region ontology.

This section on information retrieval describes commonalities among the three classifier algorithms. It is followed by a section on the fuzziness aspect of the classifications.

1) Rules. The objective is to create rules with the highest predictive accuracy for mapping independent variables (here, articles) to dependent variables (here, classification categories). Rules derived using machine learning algorithms include some that are good predictors, and some that are not, but the ones that are not good predictors are averaged with the stronger. Knowledge engineering methods, by contrast, induce rules more selectively such that each rule is likely to be a good predictor, and juggling individual rules through weighting and averaging is not crucial. Then the heuristics, or rules, are coded into algorithms using JAVA for a web-based application.

An excerpt from the time classifier appears below. The particular problem considered below is the translation of time spans from numbers to words, and how those words might lead to an article classification. "Match" of target data with the ontology results in a classification.

Convert time period numbers to words to facilitate matching. Numbers that indicate geologic time periods might appear in phrases:

[number] Ma
[number] Mya
[number] Myr (Million years ago)
[number] million years ago
[number] B.P. (before present—used for radiocarbon dating),
[number] years ago [when x is a number in the millions or uses the word million]

2) Algorithm execution. Each rule could execute in linear sequence. An advantage is that every item is classified by the rule that takes highest priority, but a disadvantage is that lower-ranked rules are harder to interpret. Alternatively, all rules could fire simultaneously. This method makes rule interpretation easier, although a higher value rule might be overlooked in favor of rule with lesser value for predicting a classification, making the classification less appropriate. We elected priority-based ordering for our research primarily because metadata location is important with respect to reliability of prediction.

3) Algorithm refinement. Analyzing mistakes made by the classifiers during the training phase and adjusting the algorithms improves their performance. That is, the algorithms may be tuned for higher classification accuracy in the general case in the hope of increasing generalizability. Algorithms are not evaluated with previously unseen documents in the testing set until after the adjustment phase is halted. Only then are the unseen items of the test set run as an evaluation. We are still adjusting our algorithms as of the writing of this paper.

4) Algorithm output. Output of classification generally has been organized into three ordered types: abstract, rank and measurement level [16]. The lowest level is abstract, providing only enough information to group items into classes. The rank level classifier groups items into an ordered list in which position reflects likelihood of belonging to the class, but there are no attached confidence values. The measurement level assigns a confidence value, however arbitrary, to each entry. Our classifiers measure at the rank level.

5) Classifier combination in future? This study involves three classifiers: one for organism name, another for time period, and a third for region. It has been found that combining classifiers may yield better performance than invoking each singly [17]. Combination in parallel tends to be employed for high accuracy, while sequential (cascaded, or vertical) combination tends to be used for speeding classification of large sets. The outcome has been found to depend not only on how the classifiers are fused, but also on how complementary they are. Our three indexes seem to provide a reasonable combination, although their optimal fusion waits for a later stage of research.

[4] http://pdftohtml.sourceforge.net

E. Classification category labels

The question for implementation was: to what level of specificity should we classify? We based on decision primarily on two factors. The first factor is database contents. We stock the database, and plan to continue to stock the database, with journals that have at least some proportion of their articles relevant to paleontology. So we assume that we will find terms that are very specific. The second factor is usability. Researchers who will be using the system presumably will have focused areas of specialization and expertise. The more precise the indexing, the better it should help our intended population of users.

We plan to populate the browse menus with those categories that appear in our result set. In this way, selecting a browse option is guaranteed to retrieve results. Not so with the keyword search, that might come yield an empty result set with error message.

F. Multi-label classification

Classification may be either single-label, also called absolute or hard, with each item classified into a single category, or multi-label, also called soft, with each item classified into one or more categories. Classification of each item by multiple categories within a facet aims to broaden item relevance to scholars. Interesting from an information retrieval perspective is less that the article has been classified into multiple categories, than that it has different levels of "belongingness" to each category.

G. Evaluation of algorithms

It has been found that desirable properties of similar algorithms include: good performance time and not overly demanding memory requirements, simple implementation of the algorithms, and robustness [18]. We propose an experiment to evaluate one aspect of the algorithms --system accuracy in categorizing and ranking journal article relevancy. We will give people the same basic rules as we give the system (classify to organism name and time period and region using the ontologies provided, and be as specific as the ontology allows). People with background in paleontology will create an answer key for a set of articles that will be considered the "ground truth" and will be assembled into a benchmark file. Then the search engine's classifications of the same articles will be compared to the benchmark. The higher the correlation between classification categories and category rankings between our system and the benchmark, the better our classifier algorithms will have performed.

IV. Fuzziness of Classification

Classifier algorithms may be designed to group items into categories whose bounds are crisp or fuzzy [19]. PaleoSearch uses three levels of fuzziness in its algorithms to show how well an item belongs to a category: a good fit (or High), an acceptable fit (or Medium), or an approximate fit (or Low). Each item may belong to as many categories as its content warrants.

What we have done is essentially to make every match between target document and ontology into a classification, and then juggle the strength of the classification with respect to the document itself. The absolute measures of High, Medium, and Low are relevancy thresholds. Our measures are based on properties of term location and repetition within the article. Within each relevancy category, the more recent a publication, the more relevant an article is assumed to be, such that the articles within High, Medium and Low sort according to publication date.

How do we know that an article is a good fit for a category and should list as High, for example? We have inspected the articles in the training set, and have formed the general rule that terms mentioned in the article title or abstract, and sometimes within the first three or so pages of the article (especially if repeated several times), are core to the topic.

The meaning of High, Med, and Low is muddied in search by keyword. This is because in keyword search, the rankings may indicate either the item's level of belongingness to a classification category, or else to its level of relevance to the keyword term entered. Consequently, the user will not know whether the classification is inexact or whether the match of retrieved article(s) to query keyword is itself inexact, or both. For example, if an article is classified as Low, but matches the query keyword exactly, it is displayed as Low. Or if an article is classified as High, but does not match the query keyword exactly, it may display as Low. Unlike in keyword search, High, Medium and Low result rankings from a browse search indicate the level of belongingness to a classification category only.

A. Example of fuzziness

To review, each article is classified into facets of organism name, time period, and region categories at a certain level of specificity. Parsed fields of article title, abstract, captions and full text are the basis of the classifications. The result is that each article is classified into multiple categories and different levels of "belongingness" (High, Med, or Low) within each facet.

The user is more interested in what matches his query than in how each article is classified. One might say that the classification of each article is the back end. For the purposes of illustration, we provide below the results of running one article through the time classifier.

This sample article[5] has been classified with highest confidence into the Mesozoic era. It has been classified with lesser confidence in two of the three periods that make up the Mesozoic—the Jurassic and the Triassic—as well as the Upper Jurassic epoch which is one of three stages of the Jurassic. It has been classified with low confidence into ages of the Hettangian within the Jurassic, and the Carnian and Norian within the Triassic.

[Title]: The coelophysoid lophostropheus airelensis, gen. nov.: a review of the systematics of "liliensternus" airelensis from the triassic-jurassic outcrops of normandy (france)

[Abstract]: —In the early 1990s a theropod dinosaur found close to the Triassic-Jurassic boundary of France was assigned to a second species of the genus Liliensternus: L. airelensis (Moon Airel Formation). This contribution reveals that common features that purportedly unite "L." airelensis with L. liliensterni are more widely distributed among coelophysoids and basal dinosaurs than it was thought. A cladistic analysis reveals that "L." airelensis is more closely related to the Coelophysidae than to L. liliensterni. A feature that supports this systematic arrangement includes a supraacetabular crest forming a well-developed ridge continuous with the lateral margin of the brevis fossa, with non-distinct notch between both structures. The new genus Lophostropheus, gen. nov., is therefore erected to include the species L. airelensis. Thus, the new combination Lophostropheus airelensis is proposed.

[Classification]: Mesozoic -- High
[Classification]: Triassic -- Mid
[Classification]: Carnian -- Low
[Classification]: Norian -- Low
[Classification]: Rhaetian -- Mid
[Classification]: Jurassic -- Mid
[Classification]: Hettangian – Low
[Classification]: Earlv Jurassic -- Mid

Figure 2 Output of PaleoSearch time classification for sample article.

B. *Visualizing fuzziness*

Visualization enables users to gain a data overview. As explained above, we use fuzz to indicate varying levels of relevance of each article either with respect to classification category, or with respect to match between query keyword and classified items (Fig. 3). We describe aspects of our visualization below. Showing fuzz is helpful because it gives the user insight into whether a retrieved result is worth a click.

"Research in visualisation of fuzzy systems is still at an early stage" [20, p.181]. No standard method is used to visualize the degree of belongingness or relative uncertainty in assigning an item to a category. It has been shown as a vertical error bar alongside each item indicating the level of accuracy, or as a sphere surrounded by a buffer zone. Another type of visualization shows box plots with distributions, or data plots in quartiles [5]. We choose not to use icons or statistical notation because they are easily misunderstood. Other possibilities include varying the size or orientation of objects, or 3D depth.

We use screen location and color to visualize relevance. It was not our intention to test every possibility for visualization, and in fact, recent tests have shown users reacting favorably to a 3D layout [4]. Our visualization is limited somewhat by what we feel are the requirements of metadata that we wish to include per article. Even now, with article title and publication date, we intend to test whether users will find even more metadata helpful in deciding whether to view a particular result.

We use a result grid of a single color whose shades indicate relative strength of connection. The color blocks are labeled "Highly relevant" for the darkest shade, "Relevant" for the middle shade, and "Somewhat Relevant" for the lightest shade (Fig. 3). We deliberately avoid labeling the lightest block "Low relevance." We do not want to mislead the user into thinking that the results are only of minor relevance. These results are just less relevant that those at the other two levels.

Studies suggest that essential to interface design is not *how* the most relevant result is shown, but *where* on screen that result appears. Users' choices of the 'best' results from different sort rankings suggest that placement on the web page (i.e., whether the result appears near the top) is most important in determining whether a given result is selected, not the actual content displayed in the top excerpts [21]. This has been called a "presentation bias" (ibid., p. 148), and we account for it in our PaleoSearch display. The best results in our grid display on the top row and spill into the second row (Fig 3). The upper left hand corner result is the very best, and relevancy reads as the language itself from left to right. Sort order within a block is determined by publication date.

Keyword searches are dogged by misspelling and inexactness in granularity with database terms. To improve the situation, we follow the Google example by giving our users a "hint" after they types a few letters into the keyword box. They might decide to select one of our default words, or else continue typing their own. Users looking for an author name, for example, will disregard our suggestions and continue with the name.

[5] M. Ezcurra and G. Cuny, G., "The Coelophysoid Lophostopheus Airelensis, Gen. Nov: A Review of the systematics of 'Liliensternus' Airelensis from the Triassic-Jurassic Outcrops of Normanday (France)," Journal of Vertebrate Paleontology, vol. 27, issue 1, 2007, pp. 73-86.

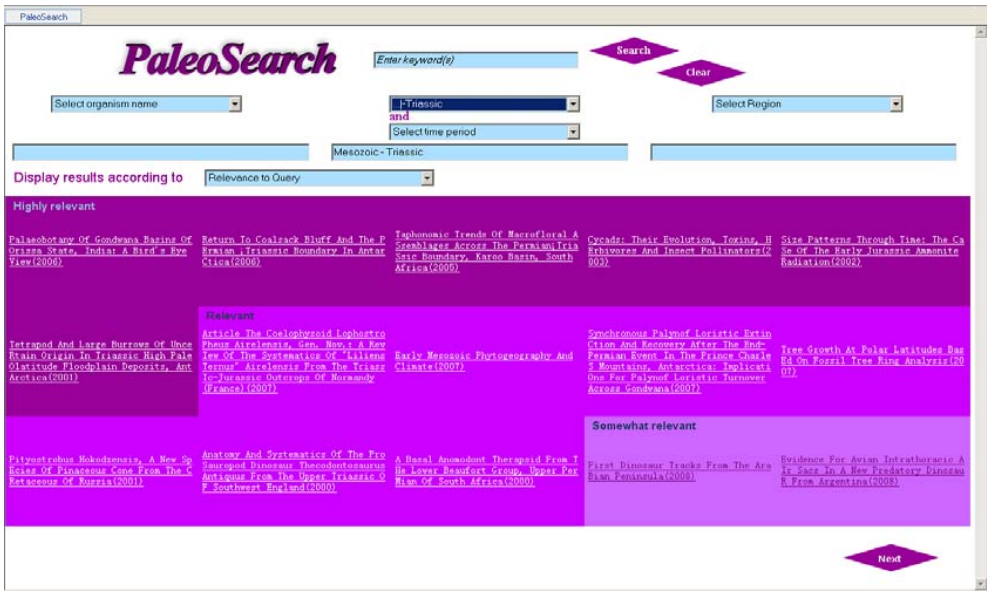

Figure 3 The interface shows three levels of relevance by labeling color block in shades of the same color with "Highly relevant", "Relevant" and "Somewhat relevant". A live version of the interface is being developed at http://paleosearch.rutgers.edu/paleo

Advantages of our fuzzy display are that it provides the user with information about each result and about the results as whole. Information about each result is shown by its position in the grid. Information about the whole result set is shown by whether the grid is full or empty. If, for example, no items are returned of high relevance but a few are low, the user knows little in the data set pertains to the query topic directly.

V. How Valuable is PaleoSearch?

We wish to determine the value of PaleoSearch to users with paleontology background. To do this, we might pose a test in which users perform a similar search task on comparable systems. The larger the number of paleontologist-participants, the more valid would be the results of our study. We will find queries that will yield comparable results both in the small collection PaleoSearch as well as in the vast digital library contenders. Participants would be asked to pose these same few questions to each system. Participants would then have a relatively even foundation to compare the systems' input mechanism, results and display. After these few tasks, we would not ask them which system they prefer, because their responses might be swayed by the Hawthorne effect (viz., trying to please investigators). Instead, we turn the question into forced choice:

> Suppose you were allowed access to only one of these systems. Which would you choose? Why?

We do not intend for our system to *replace* any other necessarily. We force a choice to suggest which system is considered most vital. The "why" request aims to get feedback on relative system strength. It would be interesting to get accuracy findings from all systems involved and see whether preference correlates with quantitative measures of system accuracy.

VI. Future Work

We will continue to collect articles from open access sources to expand the corpus, which will improve validity for both algorithm testing and to a certain extent also interface assessment. We will take articles that will work with the parser for the moment, and will improve the parser itself later. At the same time, we will be automating more steps in the process, such as the .pdf to .xml conversion necessary, and possibly also the collection of new articles.

We will continue to improve each classifier algorithm individually. When we are satisfied with preliminary results, we plan to evaluate the algorithms in the procedure outlined above by comparing system classifications and rankings to those decided by people with domain knowledge. We may also consider how to combine the classifiers for the three facets.

User testing should include determining how paleontologists determine relevance for articles in their field—or at least asking them to evaluate how well our thresholds of high, medium and low relevance correspond to what they believe to be relevance levels.

It was not our intention to compare possibilities in visualization, but only to present a workable solution for

users to try. Our visualization is limited by the comparatively large amount of metadata per result we feel is necessary to help users make judgments. Future research will involve user testing to determine the value of our system vis à vis comparable systems.

VII. Conclusion

We have developed methods to automate the parsing of .pdf journal articles into title, abstract, caption, full text, footnote and reference fields by converting from .pdf to .xml and using both layout and words within the articles to help distinguish among fields. We have developed algorithms that classify paleontology articles by organism name, time period, and region by rank-level classification. Also, we are developing an interface for the clear display of rank-level classification. Our classification and relevance algorithms are limited to paleontology. However, aspects should be generalizable and help determine relevance in scholarly articles of other disciplines, with the help of ontologies appropriate to the other disciplines. We encourage others to challenge and improve upon our work.

Acknowledgments

We are grateful for the advice of Michael Lesk from Rutgers University for all his help along the way. We appreciate the advice of Christopher Noto who convinced us of the need for a system to search around and within a range of paleontology-related articles and has been sporting in answering our paleontology questions. We thank Chaitan Baru, University of California at Santa Barbara and Director of the GEON project, for the promise of a beta test site on the Geosciences Data Network. We thank Nigel Robinson from Thomson Reuters for providing the trilobite test sample from the Index of Organism Names.

References

[1] N. D. Gershon, "Visualization of fuzzy data using generalized animation," Proceedings of the IEEE Conference on Visualization, 19-23 Oct. 1992, pp. 268–273.

[2] J. Zhai, Y. Chen, Q. Wang, M. Lv, "Fuzzy ontology models using intuitionistic fuzzy set for knowledge sharing on the semantic web," 12th International Conference on Computer Supported Cooperative Work in Design. 16-18 April 2008, pp. 465–469.

[3] R. Ji and H. Yao, "Visual and textual fusion for region retrieval: From both fuzzy matching and Bayesian reasoning aspects," MIR '07 September 28-29, 2007, Augsburg, Bavaria, Germany, pp. 159 ff.

[4] M., Deller, A. Ebert, M. Bender, S. Agne, and H. Barthel, "Preattentive visualization of information relevance," HCM '07, September 28, 2007, Augsburg, Bavaria, Germany, pp. 47–56.

[5] M. Skeels, B. Lee, G. Smith, and G. Robertson, "Revealing uncertainty for information visualization," Proceedings of the working conference on Advanced Visual Interfaces (AVI '08) 28-30 May, Napoli, Italy, pp. 376–379.

[6] T. S. Perry, "Lofti A. Zadeh (the inventor of fuzzy logic)," IEEE Spectrum vol. 32, issue 6, pp.32–35.

[7] N. J. Belkin, "Anomalous state of knowledge," in Theories of Information Behavior, K. E. Fisher, S. Erdelez and L. McKechnie, Eds. ASIST Monograph Series. Medford, NJ: Information Today, 2005, pp. 44–48.

[8] G. Salton, H., Wu, and C. T. Yu, "The measurement of term importance in automatic indexing," Journal of the American Society for Information Science, vol. 32, issue 3, 1981, pp. 175–186.

[9] P. Cimiano, Ontology learning and population from text: Algorithms, evaluation, and applications. New York: Springer, 2006.

[10] F. Sebastiani, "Machine learning in automated text categorization," ACM Computing Surveys vol. 34, issue 1, 2002, pp. 1–47.

[11] S. Purpura, and D. Hillard, "Automated classification of Congressional legislation," Proceedings of the 2006 International Conference on Digital Government Research, San Diego, California; ACM International Conference Proceeding Series; Vol. 151, pp.219–225.

[12] G. Jain, A. Ginwala, and Y. A. Aslandogan, "An approach to text classification using dimensionality reduction and combination of classifiers," Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 8-10 November, 2004, pp. 564–569.

[13] S. Kabel, R. de Hoog, B. J. Wielinga, and A. Anjewierden, "The added value of task and ontology-based markup for information retrieval," Journal of the American Society for Information Science and Technology, vol. 55 issue 4 2004, 348–362.

[14] G. Salton and M. J. McGill, Introduction to modern information retrieval. New York: McGraw–Hill Book Company, 1983.

[15] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan, "Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies," The VLDB Journal vol. 7, 1998, pp. 163–178.

[16] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," IEEE Transactions on Systems, Man and Cybernetics, vol. 23 issue 3, 1992, pp. 418–435.

[17] C.-L. Liu and H. Fujisawa, "Classification and learning methods for character recognition: Advances and remaining problems," Machine Learning in Document Analysis and Recognition, in S. Marinai and H. Fujisawa, Eds. Berlin: Springer, 2008, pp. 139–161.

[18] K. P. Bennett, K. P. and E. Parrado-Hernández, "The interplay of optimization and machine learning research," Journal of Machine Learning Research vol. 7, 2006, pp. 1265–1281.

[19] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Incremental algorithms for hierarchical classification," Journal of Machine Learning Research 2006, pp. 31–54.

[20] B. Pham and R. Brown, "Analysis of visualization requirement for fuzzy systems," Proceedings of the 1st international conference on computer graphics and interactive techniques in Australasia and South East Asia, Melbourne, Australia, 2003, pp. 181 ff.

[21] J. Bar-Ilan, K. Keenoy, M. Levene, and E. Yaari, "Presentation bias is significant in determining user preference for search results—A user study," Journal of the American Society for Information Science and Technology, vol. 60 issue 1, 2009, pp. 135–149.