Scienceography: the study of how science is written

Graham Cormode, AT&T Labs–Research
S. Muthukrishnan, Rutgers University
Jinyun Yan, Rutgers University

Abstract. Scientific literature has itself been the subject of much scientific study, for a variety of reasons: understanding how results are communicated, how ideas spread, and assessing the influence of areas or individuals. However, most prior work has focused on extracting and analyzing citation and stylistic patterns. In this work, we introduce the notion of 'scienceography', which focuses on the *writing* of science. We provide a first large scale study using data derived from the arXiv e-print repository. Crucially, our data includes the "source code" of scientific papers—the LaTeX source—which enables us to study features not present in the "final product", such as the tools used and private comments between authors. Our study identifies broad patterns and trends in two example areas—computer science and mathematics—as well as highlighting key differences in the way that science is written in these fields. Finally, we outline future directions to extend the new topic of scienceography.

1 Introduction

Many people seek to understand the progress of science by studying aspects of the process by which new scientific knowledge is created. Anecdotes and mythology abound about the process of discovery of scientific principles and design of new methodologies. Consider, for example, the narratives surrounding Newton's Theory of Gravity, or Archimedes' invention of a way to measure the volume of solid objects. Likewise, there is much study of how scientific knowledge is propagated through the scientific literature. The area of bibliometrics concerns itself with measuring properties of the research corpus, in particular, the citation patterns among texts [1, 2, 7, 8]. This leads to measures of importance, based on notions such as the citation count of a paper, the impact factor of a journal and the h-index of an author [6]. The specific application of measurement to scientific impact is known as scientometrics, and is chiefly concerned with analyzing and proposing bibliometric measures. There is also study of social aspects of science research ("sociology of scientific knowledge") and policy aspects [3].

Yet, between initial discovery and the dissemination of papers, there has been little focus on the process of *describing* scientific results, in the form of papers. While bibliometrics and sociology of sciences concern themselves with the after-effects of this work, we have relatively little insight into how the writing of science is performed. In part, this is due to the lack of visibility into this process and the intermediate steps. In a few cases, the notes and working papers of notable scientists have been made available, and these have been studied on an individual basis. But there has been no large scale study, in contrast to the analysis of citation networks containing thousands to millions of citations. Recently, there have been efforts to capture trends and influence in science, based on using both citation relations and extracted text from document collections [4, 5]. The area of quantitative data analysis also applies to track i.e. common words and bursts of interest in particular topics. Our aim is to go deeper, and learn about structures within science writing beyond the "bag of words" in each paper.

In this paper, we identify the study of this part of the scientific method as a topic of interest, which we call *scienceography* (meaning "the writing of science"). We identify a source of data that allows us to begin to measure scienceographic properties. Using this data, we are able to quantify certain key properties of science writing, its processes, and how they vary between related areas, and across time.

Our work proceeds as follows. In Section 2, we describe our data collection from the arXiv, a large collection of scientific reports. A vital property of the arXiv is that many papers are available in LaTeX format, a mark-up language that enables scienceographic study. Section 3 gives our initial analysis on two related areas, mathematics and computer science, and we compare features of the writing process. These include the use of comments to keep notes, communicate to co-authors, and adjust text; the use of additional tools such as macros and packages to facilitate the writing process; and the use of figures and theorems to illustrate the authors' intent. Finally, we conclude with directions for further study.

2 Data collection

The arXiv. Our study of scienceography was performed over the arXiv technical report service. The arXiv is an open-access web-based e-print repository that covers many scientific fields, including physics, mathematics, nonlinear sciences, computer science, quantitative biology, quantitative finance and statistics[1]. Across all areas, over 700,000 documents have been made available via the service. The service began in 1991, and is primarily maintained and operated by the Cornell University Library. After registration, users may upload new documents, or revisions of their existing documents. A distinguishing feature is that arXiv strongly encourages users to provide source files for a paper, rather than the "compiled" version. If PDF generated from TeX/LaTeX is detected, it is rejected, and the user is requested to provide source files instead.

Several formats are allowed, including TeX/LaTeX, HTML, PDF, Postscript and (MS) Word. Our study focuses on Computer Science and Mathematics where (as we see below) TeX/LaTeX predominates, and so forms the bulk of our discussion[2].

Data collection from arXiv. In addition to a conventional web interface, arXiv provides an API for access to the data[3], which we used for our data collection. Papers are arranged into a curated hierarchy: for example, `cs.AI` is the Artificial Intelligence category within Computer Science. We collected all papers with the area of computer science, and a large subset from the area of mathematics, as of April 2011. Some papers have multiple categories: a primary category, and possibly some additional categories. Our data collection method captured each paper once under its primary categorization.

As of April 2011, the arXiv listed a total of 39,015 CS papers and 196,573 Math papers under all categories, however this double counts many papers with multiple labels.

[1] `http://arxiv.org`
[2] In what follows we refer to LaTeX, with the understanding that this incorporates the TeX format.
[3] `http://arxiv.org/help/api/index`

Table 1. Dataset by filetype

(a) File Types in Arxiv

File Type	number of Papers	Ratio
pdf	7,860	12%
postscript	526	0.8%
text-html	124	0.2%
docx	151	0.2%
x-eprint	28,533	44%
x-eprint-tar	28,042	43%

(b) Filetypes by subject

	CS	Math
x-eprint-tar	14,964 (82%)	13,088 (34%)
x-eprint	3,334	25,199
Dates	1/1993 – 4/2011	1/1991 – 4/2011

We collected a total of 65,235 papers: 26,057 from CS, representing all unique papers, and 39,178 from math. For math, we picked an arbitrary subset of subcategories, and collected all papers in these categories (specifically, this was the set of subcategories ordered by their two character names in the range `math.AC` to `math.MG`).

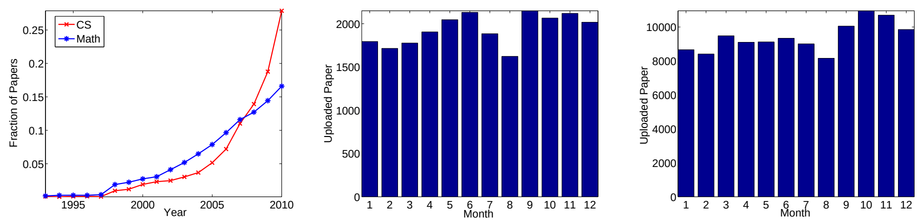
Data set. arXiv presents six fundamental document formats: the well-known portable document format (PDF) and postscript; HTML and the open XML document format used by recent versions of Microsoft Office and other wordprocessors (docx); and two variants of LaTeX, x-eprint and x-eprint-tar. Here, x-eprint corresponds to a single LaTeX source file with no other files (i.e. no additional files containing figures, bibliographical data, other LaTeX input files); while x-eprint-tar is a 'tar' archive file that contains multiple files compiled with LaTeX.

Table 1(a) shows the distribution of formats for our dataset. It is striking that within computer science and mathematics, the LaTeX formats predominate: they cover over 87% of all papers. Submissions in HTML and docx formats are negligible, totalling less than 0.4%. From the PDF files, we extracted the metadata fields of "title", "producer" and "creator". Studying these indicates that a majority of this PDF files were in fact created with the Microsoft Word software: 70% of PDFs contain "Microsoft" or "word" in these fields. We note that docx is a relatively new format, and that as of July 2011 the arXiv no longer accepts docx submissions, due to difficulties with font conversions. Instead, users of non-LaTeX tools are encouraged to submit in PDF format.

The arXiv contains papers in Math and CS going back almost two decades: papers in Math are indexed back to 1991, and in CS to 1993. Table 1(b) shows the breakdown of LaTeX types by the two major subject areas studied. There is already a striking disparity between the two styles: a majority of Math submissions are contained within a single LaTeX file, while a large majority of CS papers are spread across multiple files.

Each paper is timestamped with the date of its upload. arXiv already shows some basic statistics on month-by-month submissions for each field in its web interface[4]. Figure 1(a) shows the fraction of papers in each year for computer science and mathematics. The trend for both areas is clearly increasing over time, with an accelerating trend for CS while the growth in Math appears to be increasing linearly year-on-year. We plot the histogram for uploaded papers in each month for both subjects in Figures 1(b) and 1(c). There is a clear lull in submissions around August and July, which cor-

[4] eg, for mathematics, see `http://arxiv.org/archive/math`



(a) Papers added by year (b) CS submissions by month (c) Math submissions by month

Fig. 1. Paper submissions over time

responds to the "summer break" in many (northern hemisphere) academic institutions. We leave it to readers to conjecture explanations for this variation. Anecdotally, it is said that the summer months are used by researchers to perform new research. This may be consistent with the figures if we accept that the fruits of this research may not result in papers ready for submission until some months later. Certainly, for mathematics, October and November are months when people are most likely to submit papers to arXiv, while June and September have the highest volume of submissions in CS.

3 Structure analysis

Having access to scientific papers in LaTeX format enables us to perform analysis which is either impossible or very challenging when working with "output" formats such as PDF. For example, LaTeX files contain comments which are not present in the final output, and identify the packages (libraries) used, which is hard to do just by examining the output. We also want to study the use of expository structures like figures and theorems in scientific writing. While it is possible to identify these within PDF output, building tools to do so is difficult, due to the many ways they can appear, and the need to avoid false positives. In the LaTeX source, it is typically easier to identify these structures, since the input is plain text, and there are only a few ways to include such structures.

3.1 Comments

Much like a programming language, LaTeX allows *comments* within the source files, which are ignored by the compiler and so do not appear in the final output of the paper (such as PS or PDF). As such, they have the potential to shed extra light on the process of writing science by capturing internal communications between authors, notes, earlier drafts or excised sections. Based on our inspection of the data set, we identified the following usages of comments:

Templates and outlines. A basic use of comments is to provide an outline of the structure of the paper, either as a reminder for the authors, or to help in the use of a LaTeX template from a publisher.

Internal communication. Some comments are for communication among authors, e.g.:

```
%[xixi: Does it make sense now, as I can't find any direct reference]
```

Some authors write hints or notes in comments to remind himself/herself, e.g.,

```
%% Requires GNUPLOT, compile with  ''pdflatex --shell-escape'' for the plots.
```

Removed text. Many comments are just abandoned words, sentences or paragraphs which are removed by authors.

We begin this study by studying the prevalence and basic characteristics of comments. In LaTeX, there are a variety of methods to add comments in an article. The principle methods are:

1. the built-in latex comment command: '%'
2. use of \newcommand to define a function that ignores its parameter.
3. other more complex macros that ultimately do not produce text.
4. commands in special packages, such as packages *verbatim* and *comment*

We manually checked a large sample of papers and found the first two were by far the most common methods used. Therefore, we built scripts using regular expressions to detect their usage.

The advice on arXiv to authors uploading their papers is to remove comments from their submissions[5]. However, the above procedure found comments in 90.4% of Math papers and 95.3% of CS papers. In many cases, the comments remaining are minimal or innocuous; however we also saw many examples of the form described above, which might be considered sensitive by the authors. For CS papers, the average number of words in comments per paper is 772; for math, it was 395. Expressed as a percentage of the total length of papers, this corresponds to 7.2% in CS, and 3.9% for in Math. There is an appreciable difference in vocabulary size: in the full papers, there are around 1.3M distinct words in CS, and 1.5M in Math papers. Restricting attention to just the comments though, there are only 299K distinct words in CS papers, and 338K for Math.

We manually identify two broad clusters of terms. One cluster has terms related to mathematical expressions, such as

frac, equation, left, right, mathcal, leq, alpha, delta, sigma, phi, gamma, beta, omega, sum

The other cluster has LaTeX formatting terms, such as

figure, ldots, mbox, end, begin, label, cite, newcommand, item, section

When we compare the word frequency distribution between comments and the rest of the papers, we do not observe a very large difference. However, there are some words which are more common in comments than in the rest of papers, and vice-versa. We can find those words which have the largest absolute change in (normalized) frequency between two inputs. The 10 most discriminative words of comments in CS papers compared to the remainder of those papers, in descending order, are

latex, tex, file, use, usepackage, you, end, sty, text, version.

whereas, in the opposite direction, the top 10 discriminative words of CS papers comparing to their comments are

equation, let, each, one, def, sec, two, model, function, given.

The presence of "you" (and, more ambiguously, "use" and "version") in comments strongly suggests the importance of comments for communication between authors. In contrast, the words that are discriminative for the text seem to mostly relate to more formal computer science writing.

[5] See http://arxiv.org/help/faq/whytex

The top 10 discriminative words of comments in Math papers comparing to the rest of the papers are

tex, latex, file, end, math, macros, text, use, version, line

while the top-10 discriminative words of Math papers compared to their comments are

let, equation, such, where, theorem, proof, have, lemma, follows, proposition

which again appears to show a difference in the use of comments than for the main text.

Defining and finding comments. We have been somewhat quick in defining the concept of "comments" thus far, in the interest of adopting a workable definition for our empirical study. For a more formal notion, denote the input string as $s = s[1, n]$, where each $s[i] \in \Sigma$ for some set Σ of symbols, and assume a function (program) $P : \Sigma^* \to \Sigma^*$ that maps input strings to output strings. We can now give a semantic definition: a *comment* is a substring $s[i, j], i \leq j$, such that $P(s) = P(s[1, i-1]s[j+1, n])$. In many applications, we can assert that if $s[i, j]$ is a comment, so is a substring $s[k, l], i \leq k \leq l \leq j$. To make such comments a semantic unit, we define a *maximal comment* as a substring $s[i, j], i \leq j$, such that

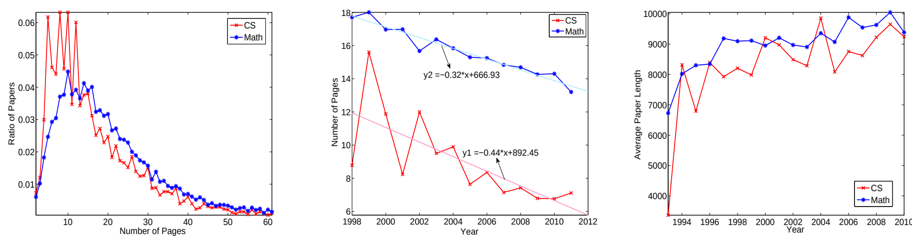
$$\begin{aligned} &P(s) = P(s[1, i-1]s[j+1, n]) \\ &P(s) \neq P(s[1, i-2]s[j+1, n]); \quad \text{and} \\ &P(s) \neq P(s[1, i-1]s[j+2, n]). \end{aligned}$$

Note that maximal comments do not overlap. Using an oracle that will check if $P(t_1) = P(t_2)$ for two strings t_1, t_2, we can now address questions of interest such as, (a) Is $s[i, j]$ a comment? (b) Is $s[i, j]$ a maximal comment, and (c) What is a partition of s into maximal comments?, and find efficient algorithms.

Mapping this problem to the LaTeX case provides further questions. In the simplest mapping, s is a LaTeX document viewed as a sequence of symbols, P is the LaTeX compiler, and the output is the pdf version (say). We assume that the output does not change when a comment is removed, and substrings $s[i, j]$ whose removal makes $s[1, i-1]s[j+1...n]$ illegal for the compiler can be detected. However, this definition means extra whitespace is treated as comments. A more LaTeX-aware way to do the mapping is to consider only parsed "words" that arise from LaTeX language, and treat them as symbols. Then the LaTeX document is viewed as the rooted *hierarchy* of environments which can be thought of as a tree. Here, the formal concepts still apply at every level of such a tree, treating symbols and nodes suitably. Finally, we can imagine simulating the LaTeX compiler, keeping its state, and detecting comments online during processing. □

3.2 Length

A fundamental property of research papers is their length: how long does it take a researcher to articulate their novel ideas? How does this vary across areas, and across time? Figure 2(a) shows the page number distribution of both subjects. The difference between the two distributions is quite striking. Math follows an approximately unimodal distribution with a peak around 10 pages. For CS, there are multiple peaks which seem to alternate page lengths. Our hypothesis is that this corresponds to submissions to conferences that had been uploaded to the arXiv: conference page limits are typically around ten pages. Indeed, the observed peaks occur at 5, 8, 10 and 12 pages, all of which are common page limits for various conferences. There is a slight preference for papers



(a) Page number distributions (b) Pagelength over time (c) Wordlength over time

Fig. 2. Length trends over time

with an even number of pages, but not excessively so: 52% of Math papers have even length, and 54%of CS papers. The average length of Math papers is slightly greater than that of CS papers: 9345 words compared to 9011 words. However, the difference in page lengths is more appreciable, averaging 15 pages in Mathematics to 9 pages in CS. This suggests a tendency to use denser page layouts in CS.

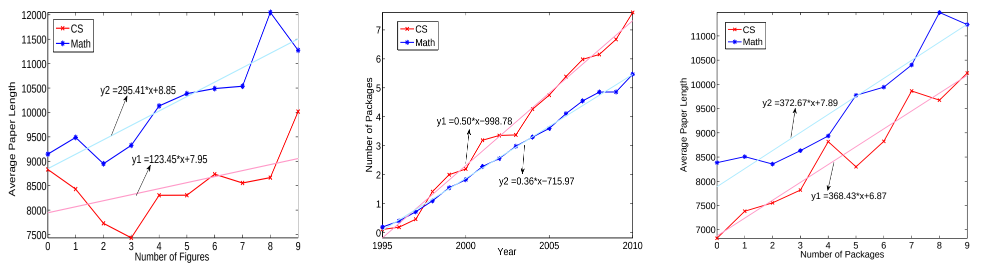
As time goes on, do papers get longer, as more concepts and related work need to be explained? The trend actually seems to be the reverse, as shown in Figure 2(b). The behavior of Math in particular seems to be well-fitted by a linear trend, removing 1/3 page per year. Extrapolating this line beyond the bounds of common sense, we conclude that the average Math paper will have no pages by the year 2052. For computer science, this date will be 2026. However, when we view the length of papers in terms of words (Figure 2(c)), we see that the trend is upwards. We conjecture the real behavior is that more papers are being posted to arXiv in dense layouts, packing more words per page.

Testing folk wisdom. The old adage, "A picture is worth a thousand words", suggests that adding illustrative figures should tend to reduce the length of a document. However, Figure 3(a) shows the opposite trend: in both Math and CS, adding figures *increases* the length of a paper. In math, the trend seems to be fairly consistent, and we have a new adage: "A pictures costs three hundred words". For CS, the trend is more variable, and weaker: the cost is an average of 120 words per figure. We might conjecture that in math, figures are typically illustrating technical concepts which require some effort to describe, whereas in CS, many figures are data plots that need less text to interpret.

3.3 Package Use

LaTeX is a very flexible and extensible typesetting system. Additional functionalities can be added by making use of "packages", via the \usepackage command. Each package implicitly captures the fact that authors need a certain kind of styling or presentation to better express their research. Hundreds of packages are available. We extracted all the packages invoked in our dataset: 1480 distinct packages in CS and 988 in Math. Tables 2 and 3 show the names of the top 20 most frequent packages in each subject, and the fraction of papers that each appears in.

American Mathematical Society (AMS) packages that provide mathematical symbols and structures like theorems (amsmath, amsymb, amsthm) are most popular. Other



(a) Paper length against figures (b) Package use over time (c) Paper length against packages

Fig. 3. Figure and Package usage

packages include figures (graphics, graphicx) and change font family and color (times, color, fontenc). These features are needed in both Math and CS. However, 'algorithm' and 'algorithmic' included by many CS papers don't appear in top 20 packages of Math. In Math, additional AMS packages are used that are not common in CS. The 10 most discriminative packages (with the largest difference in usage) for CS are

graphicx, url, epsfig, subfigure, times, color, algorithm, algorithmic, amsmath, cite

The top 10 discriminative packages for Math are

amscd, xy, amsthm, amssymb, amsfonts, eucal, mathrsfs, xypic, amsxtra, euscript

The packages discriminative for Math are all related to support for certain symbols, fonts and diagrams common in Math that are rarely used in CS. For CS, the discriminative packages cover a broader range of uses: referencing (*cite, url*), including and formatting figures (*graphicx, epsfig, subfigure*), writing pseudocode (*algorithm, algorithmic*) and styling text (*times, color*).

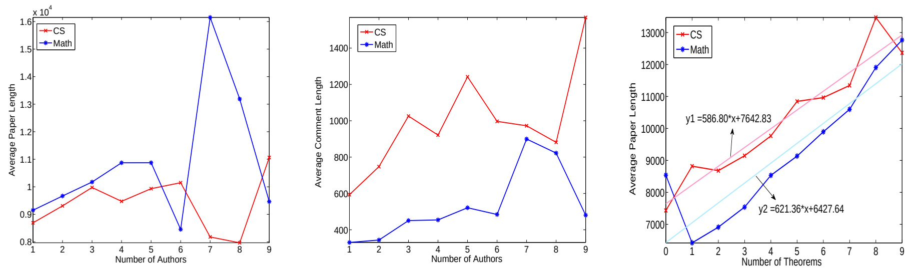
Table 2. Packages in CS

Rank	Package	Fraction
1	amsmath	0.52
2	amssymb	0.51
3	graphicx	0.50
4	amsfonts	0.22
5	epsfig	0.21
6	latexsym	0.18
7	url	0.18
8	color	0.17
9	amsthm	0.15
10	subfigure	0.13
11	times	0.12
12	inputenc	0.09
13	cite	0.08
14	algorithm	0.08
15	algorithmic	0.08
16	hyperref	0.08
17	graphics	0.07
18	fullpage	0.06
19	xspace	0.06
20	babel	0.06

In CS, 87% of papers include at least one package; while for Math, 75% of papers have packages. Of those papers which do include packages, the average number of packages included in a paper is 6.7 for CS and 5.0 for Math. These numbers are close, but indicate a slightly greater need for extra functionality in CS.

Figure 3(b) depicts how the average number of packages per paper varies by year. In math, this growth is about 1 package every 3 years, while in CS it is 1 package every 2 years. This growth rate is indicative of changing needs of authors: LaTeX is relatively stable, and rarely adds features. Yet increasingly authors need to access functionality provided by packages, such as to include URLs and graphics files in their papers.

When we plot paper length (in words) against the number of packages used, we see a different effect in Figure 3(c). There seems to be an appreciable correlation be-



(a) Words per author (b) Comments per author (c) Length against theorems

Fig. 4. Length as a function of authors and theorems

tween these two values, and moreover this is very consistent between Math and CS: each package seems to add 370 words to the paper. Perhaps a better way to view this is that as papers grow longer, they are more likely to require additional packages to help express their ideas.

3.4 Number of Authors

In some areas, it is common for multiple authors to jointly collaborate on writing a paper. Among all CS papers, the average number of authors per paper is 1.72; for Math papers, the average number of authors is 1.24. 38% of CS papers have a single author, while more than half of Math papers have just one author.

Figure 4 shows the relationship between number of authors and the length of their papers and comments, measured in words. We might expect that the length of papers should grow with the number of authors, as each author feels that they have to contribute something extra to the paper. However, we do not observe a very strong relation (Figure 4(a)): length seems to be fairly stable . For comments, there does seem to be a slight growth in the amount of comment words as the number authors rises from 1 to 2 to 3 (Figure 4(b)). So while comments may be used for discussion among authors, this does not dramatically change their length. The behavior seems more varied for more than 6 authors, but there are few papers with this many authors, so there is less support for these observations.

Table 3. Packages in Math

Rank	Package	Fraction
1	amssymb	0.57
2	amsmath	0.45
3	amsfonts	0.27
4	amsthm	0.24
5	graphicx	0.21
6	amscd	0.17
7	latexsym	0.17
8	xy	0.14
9	epsfig	0.10
10	color	0.07
11	mathrsfs	0.06
12	inputenc	0.06
13	hyperref	0.05
14	enumerate	0.05
15	babel	0.05
16	graphics	0.05
17	verbatim	0.04
18	fontenc	0.04
19	eucal	0.04
20	url	0.03

3.5 Theorems

Mathematical knowledge is typically codified in the form of theorems. Indeed, Erdös defined a mathematician as "a device for turning coffee into theorems"[6]. There are

[6] It has also been remarked that society might place greater value on a machine that works in the opposite direction.

Table 4. Comparison between Math and CS

Trends	CS	Math
submitted more than one files	84%	34%
papers with no comments	4.7%	9.6%
average number of words in comments	772	395
number of pages most papers have	6	10
papers without any packages	13%	25%
average packages included in one paper	6.7	5
papers using \newcommand	64%	66%
average \newcommand usages per paper	39.7	36.1
papers having theorems	48%	71%

many ways to define theorems in LaTeX, but for our dataset we built scripts to extract theorems based on common patterns. We confirmed that theorem use is more characteristic of math: at least 71% of Math papers contain a theorem, while only 48% of CS papers contain theorems. However, for papers with theorems, the distribution is not so different: CS papers have 4.85 theorems on average, while Math papers have 5.51. Figure 4(c) shows how paper length varies as a function of the number of theorems. Both CS and Math seem to show a similar trend, which is quite consistent: each theorem lengthens the paper by around 600 words. This makes sense: the statement, discussion and proof of a theorem should require some reasonable amount of additional text.

3.6 Comparison between Math and CS

Finally, we compare features between CS papers and math papers.

Non-textual features. Table 4 lists the key statistics that we have studied, and presents the values for each subject. While some features, such as the use of theorems and use of multiple files, are quite distinctive between the two areas, other characteristics, such as use of \newcommand are quite similar.

We performed a test of the predictiveness of these features, and built a classifier that would try to predict whether a paper belonged to Math or CS from these features alone. Using a logistic regression classifier, we were able to label 81.9% of test instances correctly. Given such a small number of features, it is perhaps surprising that the result is so accurate. Examining the parameters learned for the classifier, we saw that a lot of weight is placed on the features "new commands" and "number of theorems" to predict a Math paper. Although the likelihood of using multiple files is very different for Math and CS papers it is not significant in the classifier. Possibly this is because, while this feature is almost always 1 for CS, it is more uniformly split for Math papers.

While this showed that such features are very predictive for different subjects, the observation does not extend to sub-categories within areas: a classifier to predict which papers were in the category cs.AI (artificial intelligence) using the same set of features achieved only 57.4% accuracy.

Textual features. We compared the content words of the Math and CS papers to understand the key vocabularly difference between the two subjects. The ten most discriminative words for CS compared to Math are:

algorithm, time, figure, data, number, state, model, information, probability, problem

while the top-10 most discriminative words for Math compared to CS are:

equation, let, alpha, lambda, infty, omega, frac, gamma, mathbb, map.

While these terms should be intelligible to researchers in either field, it is clear that notions such as "data" and "information", techniques such as "probability" and "algorithm" and concerns such as "time" are central to computer science. Meanwhile, the words that define Math are mostly symbolic: "alpha", "lambda", "gamma", "omega", "infty"; or for formatting in LaTeX, like "frac" and "mathbb". Although, perhaps the best separation between the two fields comes from looking at just the most discriminating word for each: for Math this word is "equation", while for CS it is "algorithm". This seems to tally with importance of the *algorithm* package for CS noted in Section 3.3. Note that the more obvious words 'computer' and 'mathematics' do not appear in either top-10 (or, indeed, in the top-100).

4 Concluding Discussions

There has been much detailed study of individual scientists and small groups; indeed, the area of History and Philosophy of Science is based around this methodology. Yet, there has been limited large-scale study of the process of scientific communication. Primarily, this is due to the lack of available data in a format suitable for collation and analysis. Just as the growth of online social networks led to a revolution in sociology and social network analysis, so we might anticipate greater availability of scientific writing in accessible electronic form could lead to renewed interest in this area.

As mentioned in the introduction, bibliometrics and particularly citation analysis has studied in great detail how scientific papers reference each other [1, 2, 8]. Despite the size and significance of the arXiv, there has been limited prior study of this resource. For example, in 2003 the KDD conference on data mining made available 29,000 papers from the high-energy physics domain, and invited researchers to perform analysis on them[7]. However, the analysis published on this data concentrated almost exclusively on the bibliographic content of the papers, and identifying the link structure between papers, rather than any aspect of the writing style or content.

At one end of the process, there are many anecdotes about how discovery and breakthroughs occur in Science; at the other end, bibliometrics concerns itself with the aftereffects of scientific publication, via citation analysis. Between these ends of discovery and dissemination of a publication, we have relatively little insight into how the writing of science is performed and how the description of science is compiled. We have identified the study of this part of the scientific method as a topic of interest and coined the term *scienceography* (meaning "the writing of science") to frame the area.

In the past, there has been very little visibility into this aspect, but we have made a case that with the availability of LaTeX source in arXiv together with the timestamp, we have a data source where certain basic aspects of scienceography can be studied. There is much more to be done expanding the empirical studies in Scienceography, as well as identifying the basic principles and developing a theory of Scienceography.

[7] See `http://www.cs.cornell.edu/projects/kddcup/datasets.html`

Expanding Empirical Studies. Getting access to version control information used in writing papers can provide more insights into how research papers are composed[8]. For example, it is common to expect that papers are produced in various sections (perhaps by different authors) and then combined with various "passes" by different authors. Does the data validate this model? There are portions that are written and then removed from final publication. Can we examine the intermediate forms of a research paper and its evolution over time? At a more detailed level, can we quantify the "effort" (in terms of time and author hours) needed for producing portions of the paper, and indeed predict time needed from current state to the final state, given the portions that need to be generated (and predict the probability of making a deadline for a conference or a grant proposal)? Going beyond research papers, we can consider research presentations. There is data on the web which not only consists of the powerpoint slides, but also "comments" in the form of author notes for each slide which are not visible to the audience during the presentation. What insights can these provide into delivery of research results by speakers?

Building a Theory of Scienceography. There are basic questions about models at the macro level of communities as well as at the micro level of individuals and individual papers. For example, at the macro level, can we develop models for how writing styles and norms (say use of packages, naming methods for theorems or figures, and others) migrate from community to community? Can we model the time dependence of how research progresses (as seen by uploaded publications) over time in different communities? At the micro level, are there models of social interactions of authors that can predict the salient—scienceographic—features of a paper? □

Acknowledgements. This material is based upon work supported by the National Science Foundation Under Grant No. 0916782.

References

1. H. B. Atkins and B. Cronin. *The Web of Knowledge*. Information Today, 2000.
2. N. D. Bellis. *Bibliometrics and Citation Analysis: From the Science Citation Index to Cybermetrics*. Scarecrow Press, 2009.
3. K. H. Fealing, J. I. Lane, J. H. M. III, and S. S. Shipp, editors. *The Science of Science Policy: A Handbook*. Stanford University press, 2011.
4. S. M. Gerrish and D. Blei. A language-based approach to measuring scholarly impact. In *International Conference on Machine Learning*, 2010.
5. G. Goth. The science of better science. *Communications of the ACM*, 2012.
6. J. E. Hirsch. An index to quantify an individual's scientific research output. *PNAS*, 102(46):16569–16572, 2005.
7. S. Klink, P. Reuther, A. Weber, B. Walter, and M. Ley. Analysing social networks within bibliographical data. In *DEXA*, pages 234–243, 2006.
8. H. F. Moed. *Citation Analysis in Research Evaluation*. Springer, 2011.

[8] Similar analyses have been performed on open source code, such as the Linux kernel, `http://www.vidarholen.net/contents/wordcount/`