

# How to Cheat the Page Limit: the Cheating Version

Wouter Duivesteijn<sup>1,2</sup> (✉), Sibylle Hess<sup>1,2</sup>, and Xin Du<sup>1,2</sup>

<sup>1</sup> Proceedings Chairs of ECMLPKDD 2019

<sup>2</sup> Data Mining Group, Technische Universiteit Eindhoven, Eindhoven, the Netherlands,  
{w.duivesteijn,s.c.hess,x.du}@tue.nl

**Abstract.** Every conference imposing a limit on the length of submissions must deal with the problem of page limit cheating: authors tweaking the parameters of the game such that they can squeeze more content into their paper. We claim that this problem is endemic, although we lack the data to formally prove this. Instead, this paper provides a far from exhaustive summary of ways to cheat the page limit, a case study involving the papers accepted for the Research and Applied Data Science tracks at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD) 2019, and a discussion of ways for program chairs to tackle this problem. Of the 130 accepted papers in these two ECMLPKDD 2019 tracks, 68 satisfied the page limit; 62 (47.7%) turned out to spill over the page limit, by up to as much as 50%. To misappropriate a phrase from Darrell Huff’s “How to Lie with Statistics” [3], we intend for this paper not to be a manual for swindlers; instead, nefarious paper authors already know these tricks, and honest program chairs must learn them in self-defense.

**Keywords:** Scientific integrity · Conference organization · Reviewing process · L<sup>A</sup>T<sub>E</sub>X · Program chairs

## 1 How to Cheat the Page Limit: Authors’ Viewpoint

So you’re a scientist, you have a good idea, and hence you are preparing a paper to submit to a big international conference. For instance, you’re a data miner, and you have set your sights on the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases [2] (ECMLPKDD). In many scientific disciplines, this compels you to prepare your manuscript in L<sup>A</sup>T<sub>E</sub>X, and the conference will have outlined a set of rules along which you are supposed to do so. For instance, ECMLPKDD publishes their accepted papers in Springer’s *Lecture Notes in Computer Science*, and directs you for more information to the corresponding web page [6]. The conference website informs you that within this format, the maximum length of papers is 16 pages, including references. Armed with this knowledge, you download a copy of the Guidelines for Proceedings Authors [5], and you commence writing!

Suppose that, once you emerge from your burst of creative writing and once the dust settles, you run into a problem: the paper you produced is actually 18 pages long instead of the allowed 16. This is a bummer. At this point, the appropriate path to take, the high road, is to reformulate your thoughts and arguments in a more concise manner. Remove sentences, reformulate arguments such that they take up less space, consider whether this train of thought is really necessary for the overall story line. There are two problems with this solution. On the one hand, it is a lot of work. On the other hand, if you try to convey your thoughts while being more economic with the number of words you use, you run the risk of making your paper more difficult to read. This would put you at a disadvantage, compared to those authors that use nefarious tricks to squeeze their 18-page paper into 16 pages. Fear not! You too can learn these tricks!

### 1.1 Load Alternative Fonts

Springer prescribes the use of L<sup>A</sup>T<sub>E</sub>X’s standard Computer Modern font [5, Section 2.3]. However, this font eats loads of space. If one were to load the Times font, with `\usepackage{times}`, this will easily eat a page out of your paper.

For advanced users, one might want to load the packages `helvet` and `courier` along with `times`, so that you have the plausible deniability that you’re really only doing this for aesthetic reasons. Also, if you fear that the `times` package is too obvious, you might instead want to use either `\usepackage{newtxtext,newtxmath}` or `\usepackage{stix}`.

A drawback of this method is that loading alternative fonts is really obvious: a reviewer might actually catch you doing this, and a small subset of those reviewers might elect to reject your paper over it. This risk may be too much for you to take.

### 1.2 Reduce Font Size

Springer prescribes that all plain text is written in 10pt format, except the abstract and captions (both of which may be written in 9pt format) and section headers (which are bigger) [5, abstract and Section 2.1]. To get around these



**Fig. 1.** This portrait is fictitious. No identification with actual persons (living or deceased), especially any ECMLPKDD 2019 chairs, is intended or should be inferred.

limitations, many font size altering L<sup>A</sup>T<sub>E</sub>X commands are at your disposal: `\small`, `\smaller`, `\tiny`, `\footnotesize`, `\scriptsize`, and `\fontsize{7pt}{7.5pt}` are some examples. If you put one of those in a mathematical equation, small gains are made; if you put a `\scriptsize` inside a table, you can gain significant space!

For advanced users, one might want to employ a `\scriptsize` inside the `\lstset` settings of the `listings` package. This is used to control the appearance of code in your paper, and the font size change is not that obvious even to the most accurate of proceedings chairs. Alternatively, one could pass the font size changing commands as options to the `caption` package (where `\small` would be allowed, but none of the others), or use them to change the appearance of `algorithm` environments by changing `\SetAlFnt` or `\SetCommentSty`.

### 1.3 Reduce Vertical Spacing...

Springer does not allow authors to insert or remove vertical spacing [5, Section 4.1]. However, if you're not too aggressive with it, it is unlikely that reviewers will catch you doing so. Lots of whitespace is left around section headers in the standard Springer style, which you might want to reduce by `\vspace{-1cm}` or `somesuch`. If you're really old school, you might want to use the plain T<sub>E</sub>X command `\vskip -1cm`. If you need to reduce the effect of a single newline, a `\[-3mm]` might just give you the subtle edge you need without being too obvious.

**...Around Equations** If you feel that the mathematical equations get too much whitespace, one can set some flexible spaces to narrower amounts. With `\abovedisplayskip` you can control the amount of whitespace before an equation, and `\belowdisplayskip` controls the whitespace after an equation.

**...Around Algorithms** To reduce whitespace around algorithms, set `\algomargin` to a smaller value.

**...in and Around the Bibliography** To reduce whitespace between entries of the bibliography, one can set `\bibsep` to a smaller length. Additionally, for the true hackers, the command `\bibpreamble` is meant to print some text at the start of the bibliography; you can abuse it to provide some additional negative space commands.

**...Between Items** If one uses `itemize`, `enumerate`, or `description` environments, one can reduce the spacing between individual items with the `\itemsep` length.

**...Throughout the Document** For advanced users, you might not want to keep typing negative vspace everywhere in the document. There is a neat trick that allows you to reduce vertical spacing between all lines in one single command: just set `\linespread{0.9}`. This does change the physical appearance of the document quite a bit, so it is a bit risky. Renewing the `\baselinestretch` command to a lower factor has a similar effect.

### 1.4 Reduce Spacing Around Figures

Many academic papers contain figures, not entirely dissimilar from Figure 1. Springer guidelines leave authors relatively free in designing the contents of figures. However, the standard style does prescribe the spacing of these figures, and it eats quite a bit of space. We could tackle this with negative `\vspace`s, but subtler ways are available. You can control the distance between floats (figures, tables, algorithms, ...) on the top (or bottom) of the page and the text by setting the length `\textfloatsep`. If several floats follow one another, you can control the distance between them with `\floatsep`. If floats are inserted inside the page text (using the `h` placement option), the distance to the surrounding text is controlled with `\intertextsep`.

If you have a figure as narrow as Figure 1, you might be annoyed by the waste of horizontal space. If you have multiple figures, of course, you can combine them into one using the relevant  $\LaTeX$  packages, but this document only has a single version. The standard  $\LaTeX$  solution for this is to employ the `wrapfig` package. This package enables the figure to be on either the left or the right side of the page, while the normal text flows on alongside it. The Springer guidelines are unclear on whether or not this is acceptable. One could see the use of `wrapfig` as a temporary edit to the page margin, which surely is not allowed, but this is a matter of opinion. If you use `wrapfig`, and use the vanilla `wrapfigure` environment, you will end up with comically large vertical spacing surrounding your figure, and you might be tempted to counter those with negative `vspaces`. While the Springer guidelines forbid inserting or removing vertical spacing, this particular spacing is inserted by the `wrapfig` package, which makes removing the spacing you just inserted a knotty argument to disentangle. The question whether `wrapfig` is compatible with the Springer guidelines is unclear at the time of writing; it would be prudent for Springer to provide clarity in this matter.

## 1.5 Reduce Table Space

You can reduce the spacing *around* tables in the same way as the spacing around figures, as discussed in the previous section. However, there are additional cheats involving tables, which would not be considered cheats when they would involve figures. The core point is that tables are supposed to display plain text (typically, numbers), for which Springer prescribes a standard font size (10pt). Any reduction of this font size, be it implicit or explicit, violates the Springer guidelines.

If you're a data miner, chances are that you want to display the dominance of your method over the competition. The more competitors you beat on the more dataset, the better! Hence, many data mining paper have at least one quite large result table. If your table is not entirely square, you have the choice of mode. A table in portrait mode is more likely to fit within the margins, but it eats a lot of vertical space. A table in landscape mode is more economical in space, but it more likely to walk outside of a page margin. If this happens, you can reduce the horizontal size of your table with `\resizebox`, `\scalebox`, or `\adjustbox`. Collateral benefit is that this often also reduces the vertical size of your table, which gives you more lines of text to write your arguments. The implicit reduction of table content font size violates the Springer guidelines<sup>3</sup>.

If you wish to reduce the row height of tables, you can set `\arraystretch` to a lower factor. Since this inserts or removes vertical spacing, this violates Springer's guidelines. If you wish to reduce the space between table columns, you can reduce the setting of `\tabcolsep`. Since this does not directly affect vertical spacing, its compliance with Springer's guidelines is unclear; proceed with caution.

For advanced users, you can get around all these hacks by creating your table in an external piece of software (Excel, or Paint if you want to be really creative), turning it into a figure file (png, pdf, jpg, ...), and then importing that file as a figure. This makes it prohibitively time-consuming for the proceedings chairs to recreate your table as a standard  $\LaTeX$  table, while also making it impossible for the proceedings chairs to check the font size in your pictured table.

## 1.6 Reduce Margins

The Springer standard style leaves huge white margins around the edges of the paper. If you want to make more economical use of the pages, you might want to reduce those margins. If you want to be really brazen about this, use `\usepackage[margin=1in]{geometry}`. Slightly less nuclear options are `\usepackage{fullpage}` and `\usepackage{a4wide}`. Only use this if preceding cheats in this section do not deliver the desired results! Changing the margins is really visible in the final papers, and if a reviewer catches it, this is sufficient grounds to desk reject your paper.

For advanced users, if you only need to steal the bottom margin on a single page, you could employ `\enlargethispage{1cm}`. A similar effect can be achieved by collecting multiple figures that together would not fit within the margins of a single page, and put them into a single float environment.

## 2 Case Study: the 130 Papers Accepted at the Research and Applied Data Science Tracks of ECMLPKDD 2019

We have no hard data across multiple conferences or venues to say anything substantial about how widespread this phenomenon is. However, in our role as Proceedings Chairs at ECMLPKDD 2019 [2], we have acquired the material to analyze the extent of space cheating in a single edition of a single conference. At this conference, 130 papers were accepted for publication in the Research and Applied Data Science tracks. All authors of these 130 papers were required

<sup>3</sup> Notice that these three kinds of boxes are compatible with the Springer guidelines when used to resize figures! This is often even necessary to allow a figure to fit on a page. Conversely, resizing tables or algorithms thusly runs afoul of font size specifications, and hence violates the guidelines.

**Table 1.** Histogram of paper lengths in pages (partially) used, across the 130 papers accepted for the Research and Applied Data Science tracks at ECMLPKDD 2019. Notice that the page limit is 16 pages. The second row contains the histogram of the papers compiled as they were originally submitted; the third row contains the histogram of the papers recompiled after all space cheating commands from Section 1 were removed.

number of pages	12	13	14	15	16	17	18	19	20	21	22	23	24
submitted version	1	0	2	5	112	10	0	0	0	0	0	0	0
reformatted version	1	0	1	5	61	49	8	3	1	0	0	0	1

**Table 2.** Condensed form of Table 1, retaining only the information whether papers were over or under the limit.

complied with page limit?	yes	no
submitted version	120	10
reformatted version	68	62

to send us their  $\LaTeX$  sources, so that we could prepare the pre-proceedings versions of their paper, including the Springer-style headers and such. With those  $\LaTeX$  sources, we did the following.

First, we compiled the sources as the authors delivered them to us. This gives us the page length of the papers as the authors intended, on our system. Subsequently, we removed all commands listed as space cheating in Section 1. This gives us the page length of the papers, in the form compliant with Springer’s guidelines [5].

## 2.1 Results

With the sole exception of `\enlargethispage`, all forms of space cheating mentioned in Section 1 appeared in at least one of the 130 papers.

Paper length histograms are given in Table 1, both in the form as originally submitted by the authors, as in the reformatted form. In Table 2, we summarize for both versions how many papers complied with the limit.

The first thing to notice is that even in the originally submitted version, 10 of the 130 papers do not comply with the page limit (cf. Table 2). This may be an artefact of people using old versions of the LNCS  $\LaTeX$  class and the associated bibliography style. It may also result from small discrepancies in  $\LaTeX$  distributions across machines and operating systems. Most of these ten are not necessarily nefarious.

The second thing to notice is that with all space cheating removed, 62 out of 130 papers are over the page limit (cf. Table 2). A whopping 47.7% of papers are overlong. There is a huge variety in the degree to which papers are overlong. The vast majority (49 out of 62) of these papers roll over onto page 17 (cf. Table 1) and that’s it. Some of these papers have only a single reference spill over on page 17, making the space cheating rather benign, while other make use of the entire 17<sup>th</sup> page. It is clear, though, that this form of space cheating is on another level than the paper that manages to squeeze a 24-page paper (when complying with the guidelines) into a 15-page submission (when employing tricks from Section 1).

Finally, we would like to remark that of the papers remaining on the correct side of the page limit in their reformatted version, not all are innocent either. Only 31 of the 130 papers did not contain any of the space cheats from Section 1. Two additional papers had a single negative `\vspace`; all others had more serious space cheats. However, for 35 papers, the space cheating did not result in a change in the number of pages the paper used, which is why we do end up with 68 papers complying with the page limit in reformatted form.

## 3 How to Tackle the Problem

In this digital day and age, you might wonder why we even bother with page limits to begin with. Wouldn’t it be a better idea to just let authors write as much as they want, and relegate this discussion to the dustbin of history along with the concept of an actual physical proceedings book? In smaller research fields, this may be a very reasonable solution to the situation sketched so far in this paper. The problem<sup>4</sup> in data mining is that the field is currently going through such a boom, that reviewers are severely overstretched. Collectively, we can barely handle the reviewing load as it is, and if we wouldn’t keep papers that are to be reviewed relatively short, we would move the reviewing load one order of magnitude even further away from sustainable. Hence, we don’t see the page limit disappear any time soon.

As long as we still have a page limit, you might wonder if this is really such a big deal. Do we really need to be so strict with people skirting around the margins? The thing is, as we outlined in the second paragraph of Section 1, when writing an overlong paper, the high road would be to be more economic with your phrasing. When reducing

<sup>4</sup> We call it a problem because it poses a problem for the particular situation sketched in this paper. Outside of this particular context, experiencing a boom in your research field can of course be quite beneficial.

the overlong paper to the page limit, when authors take the high road, they will cut corners in the number of words, sentences, arguments used to fortify their case. Less fortification for the paper’s arguments reduces the probability that the paper is accepted. This is not a problem on its own, *as long as the playing field is level*. If other overlong papers get to use tricks from Section 1 to squeeze under the page limit, and if those tricks are not punished, they do not suffer this reduced probability. Hence, if we do not counter page limit cheating tricks, we put authors taking the high road at a disadvantage, which is not a signal we as a community should want to broadcast.

Therefore, we say: when a page limit is in place, authors should be forced to comply with it following the accompanying guidelines as strictly as humanly possible.

### 3.1 Benefits and Limitations of Automation in this Process

The contents of this paper are the direct result of lots of manual work on behalf of the proceedings chairs. For a conference of the magnitude of ECMLPKDD, this is still doable; if one were to replicate this analysis for a conference of the magnitude of AAAI 2019, where 1150 papers were accepted, it is a ridiculous amount of work. Beyond that, not all proceedings chairs may feel strongly enough about this topic to invest the time necessary for this analysis. However, the job does seem like a good candidate for outsourcing to automation.

Springer’s Data Mining and Knowledge Discovery journal, for instance, has an online system for paper submission enforcing style guide matters. When submitting a paper to the journal, you must upload your  $\LaTeX$  sources to their website, and they will compile a submission PDF for you from your sources. This gives them a lot of control on what happens within those source files. Couldn’t we use a similar system to solve our problem?

**System Overload** The problem is that the situation for journals and conferences are not identical. For relatively small scientific communities, the answer may very well be positive. However, as mentioned before, we are data miners, and data mining is booming. So booming, in fact, that the sheer volume of paper submissions managed to break the Microsoft Conference Management Toolkit around the paper submission deadline of NeurIPS 2019 [4]. The simple confluence of thousands of papers attracted by a submission deadline manages to break our electronic systems already; if we were to add to this process an automated  $\LaTeX$  compilation from source files, this would inflate this problem to an unacceptable degree. Hence, we cannot make automated source compilation from scratch part of the paper submission process.

We can, however, demand that authors upload a zip file with their source material to the existing systems instead of their self-compiled PDF, and analyze the contents of that zip directly after the paper submission deadline.

**The Problem with Automated Desk Rejects** Another problem is that we must keep the human in the loop. Several of the commands we highlight in Section 1 can be used for both nefarious and perfectly legal means. For instance, `\resizebox`ing a figure is perfectly acceptable, while `\resizebox`ing a table illegally reduces the font size. People can reduce `\arraystretch` to cheat space, but they can also increase `\arraystretch` simply to make their table look a bit better. We cannot simply automatically detect some commands and reject the paper using them without seeing the papers. Instead, the best we can do is run an automated check on whether any of the commands from Section 1 appear in the  $\LaTeX$  sources, and if so, flag up the paper for manual inspection by the proceedings chairs.

### 3.2 Our Proposal: the $\TeX$ nical Desk Reject Phase

We propose the following change in the paper submission and review procedure at ECMLPKDD, starting at the 2020 edition. After the paper submission deadline, plan a week for the  $\TeX$ nical Desk Reject Phase. It works as follows.

1. As usual, paper bidding starts as soon as the abstract submission deadline is over. Papers are assigned to reviewers as usual.
2. For the paper submission deadline, authors submit a zip file with their  $\LaTeX$  sources, along with their self-compiled PDF. The author-submitted PDFs are saved for dissemination to reviewers.
3. Directly after the paper submission deadline, one of the proceedings chairs runs the submitted zip files through an automated system, to be designed. The only functionality is that the system checks whether any of the commands listed in Section 1 appear in the submitted  $\LaTeX$  sources. If at least one of the commands appears, the paper gets flagged.
4. In the week after this automated check, the proceedings chairs recompile all the flagged papers, removing all commands from Section 1 that are deemed offending (keeping the distinctions of Section 3.1 in mind). If the recompiled version complies with the page limit, the flag is removed. If the recompiled version violates the page limit, the paper is  $\TeX$ nically Desk Rejected.
5. The program chairs inform the authors of  $\TeX$ nically Desk Rejected papers of the decision.

6. The program chairs inform the reviewers having  $\text{\TeX}$ nically Desk Rejected papers in their batch of the decision, pointing out that their review for this paper is no longer required.

A precondition for this procedure to work smoothly, is that we must clearly communicate to all authors that we plan to include this phase, and under which conditions their papers will be  $\text{\TeX}$ nically Desk Rejected. The extensive list of examples of what we consider page cheating as listed in Section 1 should be disseminated, so that no-one can complain that they didn't know what is or is not illegal.

## 4 Illustrative Example

To illustrate the extent to which space cheating can distort the employed number of pages, we have generated two versions of this paper: the Version Adhering to the Guidelines [1], and the Cheating Version (which you are currently reading). Beyond some small textual differences in this section, the only differences between the two are that we added the following commands to the Cheating Version:

1. `\usepackage{stix};`
2. `\linespread{0.9};`
3. `\usepackage[margin=1in]{geometry}`

As a result of these changes (which we lifted verbatim from the 130 submitted papers), the Version Adhering to the Guidelines is 10 pages long, but the Cheating Version is merely 6 pages.

## 5 Conclusions

$\text{\LaTeX}$  offers far too many ways to change the appearance of a paper. As a consequence, it is far too easy for authors to squeeze in more material than the page limit would allow if one were to play the game fairly. We outline several such ways to cheat the page limit in Section 1, and illustrate its effect on the accepted papers in the Research and Applied Data Science track at ECMLPKDD 2019. As we have seen in Table 2, 47.7% of the papers violate the page limit when recompiled according to the Springer guidelines, with the biggest offender adding a whopping 50% extra pages (cf. Table 1). To tackle the problem, we propose to include a  $\text{\TeX}$ nical Desk Reject Phase into the process, directly after the paper submission deadline.

While this paper is largely written from the viewpoint of data miners and a specific data mining conference, we think that its lessons can be generalized beyond this scope to any venue where submissions have to comply with a page limit. We think that the automated system required to support proceedings chairs in completing the new phase will not be too difficult to build, yet it ought to substantially reduce the workload involved in performing this task. When done well, the  $\text{\TeX}$ nical Desk Reject Phase will result in more fairness in the conference submission evaluation procedure, by being harsher towards papers not playing by the rules.

## References

1. Wouter Duivesteijn, Sibylle Hess, Xin Du: *How to Cheat the Page Limit: the Version Adhering to the Guidelines*. To appear.
2. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases: *ECMLPKDD 2019*. Available online: <http://www.ecmlpkdd2019.org/>. Accessed August 15, 2019.
3. Darrell Huff, Irving Geis: *How to Lie with Statistics*. Victor Gollancz Limited, London, 1954.
4. Tony Peng: *Paper Submissions Break NeurIPS 2019 Paper Submission System*. Available online: <https://medium.com/syncedreview/paper-submissions-break-neurips-2019-paper-submission-system-884a60e32a82>. Accessed August 15, 2019.
5. Springer: *Guidelines for Proceedings Authors*. Available online: [ftp://ftp.springernature.com/cs-proceeding/svproc/guidelines/Springer\\_Guidelines\\_for\\_Authors\\_of\\_Proceedings\\_CS.pdf](ftp://ftp.springernature.com/cs-proceeding/svproc/guidelines/Springer_Guidelines_for_Authors_of_Proceedings_CS.pdf). Accessed August 15, 2019.
6. Springer: *Information for Authors of Springer Computer Science Proceedings*. Available online: <https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines>. Accessed August 15, 2019.