

LaTeX入門

Overleafでレポート作成

正田 備也

masada@rikkyo.ac.jp

参考資料

- TeX/LaTeX入門
 - <https://ja.wikibooks.org/wiki/TeX/LaTeX%E5%85%A5%E9%96%80>
- TeX Wiki
 - <https://texwiki.texjp.org/>
 - LaTeX入門
 - <https://texwiki.texjp.org/?LaTeX%E5%85%A5%E9%96%80>
- Overleaf documentation → これを主に参考にしています
 - <https://www.overleaf.com/learn/>

論文は、普通、LaTeXで書く（？）

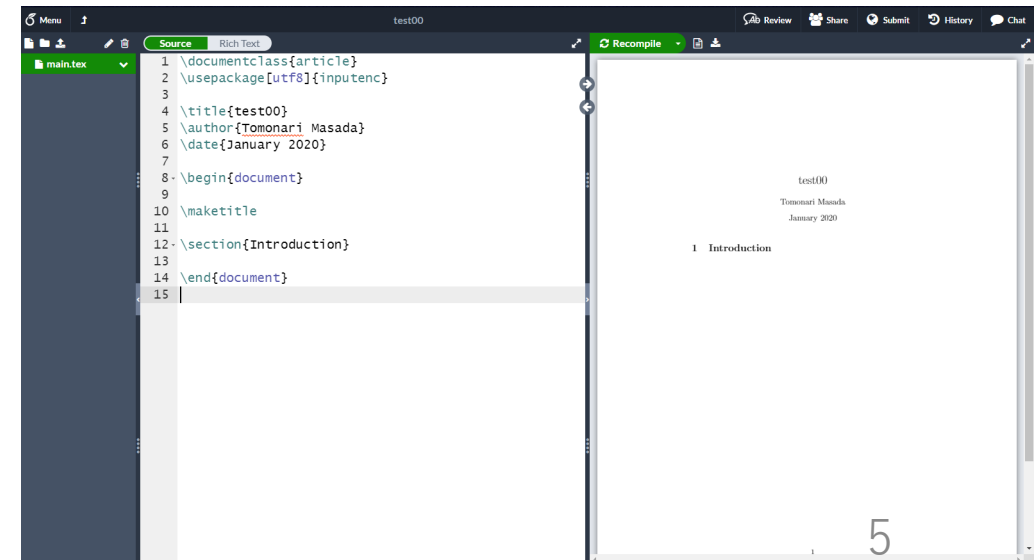
- 英語の場合、論文をWordで書くことはあまりない
- 国際的に活動したい人は、LaTeXで文書作成できるようにしておくこと
- 必要最低限のことを知っておけばよい（あとは必要に応じてググる）
- 「LaTeX」は英語圏では「レイテック」と読む人がほとんど
 - 「レイテックス」もOK
- 最近はLaTeXの環境を手元に用意せずWeb上のサービスを使うことが多い
- 代表的なサービスがOverleaf

今回は説明しないこと

- Overleafそのものの細かい操作方法是各自調べてください
- Overleafで日本語をどう使うかは各自調べてください
 - Overleafで論文を書くときはたいてい英語だと思いますので🙄
 - <https://ja.overleaf.com/learn/latex/Japanese>
 - <https://scrapbox.io/shoji-lab-survey/OverLeaf%E3%81%A7%E6%97%A5%E6%9C%AC%E8%AA%9E%E3%81%AE%E8%AB%96%E6%96%87%E3%82%92%E6%9B%B8%E3%81%84%E3%81%A6%E3%81%BF%E3%82%88%E3%81%86%E3%BC%81>
- 手許のマシンにLaTeX環境を構築する方法も、説明しません
 - 個別の環境に依存しますので・・・各自調べてください

Overleafで新規プロジェクトを作る

- 「New Project」をクリック
- 今回は「Blank Project」をクリック
- 左半分がソース
- 右半分がコンパイル後のPDF
- LaTeXでの文書作成は・・・
 - 左半分のソースを編集し・・・
 - それをコンパイルしてPDFにする
 - コンパイルは「Ctrl + Enter」



MacでOverleafを使う時の注意

- カーソルの位置が、実際に文字を入力する位置からズレる
- 対処法：フォントを等幅フォントにする
 - 左上の「Menu」をクリック、
 - 「Font Family」で「Monaco」を選択

課題3 (今日は最初に課題を言っておきます)

- GANの原論文をarXivからダウンロードする
 - <https://arxiv.org/abs/1406.2661>
- 論文のSection 4の4.1までと、Section 5とを、そのまま"写経"する
 - 自分が論文を書くときに、他人の論文をこのように丸写しするのは、当然ダメです！
 - 従来研究の紹介は、元の論文の英文を必ずパラフレイズすること！
 - 自分の過去の研究の紹介であっても必ずパラフレイズすること！
 - Webのパラフレイズサービスは当てにならないので使わないこと！
 - 今回はあくまでもLaTeXの練習なので、丸写しします
 - その範囲に出てくる参考文献も含めて丸写ししてください
 - 文献番号は違っていてもよいです
- ソースをzipでかためたものと、PDFとを、提出してください

LaTeXの基本

LaTeXでの文書作成の考え方

- ソースファイルをコンパイルしてPDFファイルを作成する
- 地のテキスト以外に、コマンドを多用する
- コマンドで、文書の見たい目や構造をコントロールする
- コマンドは「¥」で始まる
 - 「¥コマンド{引数}」という種類のコマンドや・・・
 - 「¥begin{環境名}」～「¥end{環境名}」という種類のコマンドがある
 - 囲まれた部分を特定の環境として設定する

* TeXとLaTeXの違い

- TeXは1978年にDonald E. Knuthによって開発された組版エンジン
 - TeX言語というマークアップ言語を使って、文章とその構造や装飾に関する付加情報とを、テキストファイルの中に書きこむ
 - そのテキストファイルをTeX処理系に入力して、印刷用の紙面を得る
- しかし、TeX言語で自分が思うレイアウトの文書を作るのは大変
 - 文書の構造や装飾の指定に使う命令が、あまりにもプリミティブ
- そこで、Leslie Lamportという人が1985年にLaTeXを作った
 - LaTeXは、マクロの体系
 - マクロ=TeXのプリミティブな命令を組み合わせて、文書の構造や装飾について誰もが使いそうな高機能な命令を記述したもの

¥documentclass[]{}

- 文書全体のレイアウトを定義
- 定義に使うクラスファイル（拡張子が`.cls`）のファイル名を波括弧の中に書く
- 論文投稿時には、投稿規定によってどのようなクラスファイルを使うか指定されているので、それを使えばいい
 - 何でもいいときは、よく¥documentclass{article}を使う
- 角括弧の中にはオプションを書く
 - 省略するとデフォルトのオプションが使われる
 - 用紙のサイズや基準となるフォントサイズを指定する
 - a4paper, a5paper, b4paper, b5paper
 - 10pt, 11pt, 12pt
 - などなど

プリアンブル

- LaTeXのソースファイルのなかで、どのパッケージを使うかを書いたり、独自のマクロを定義するために使う場所
- だいたい`¥documentclass{}`から`¥begin{document}`の間
- どのパッケージを使うかは`¥usepackage[]{}コマン`ドで書く
- 独自のマクロは`¥newcommandコマン`ドで書く

¥usepackage[]{}

- 波括弧内のファイル名で指定されたパッケージを使う
 - 角括弧内はそのパッケージ用のオプションを書く
- 例) ¥usepackage[noend]{algpseudocode}
- パッケージは別のTeX文書で、初期状態では用意されていない機能を追加できる
- よく使うパッケージにamsmath, url, algpseudocodeなどがある
 - 欲しい機能を持つパッケージをググって探せばよい
- Overleaf : What packages do you support?
 - https://www.overleaf.com/learn/how-to/What_packages_do_you_support%3F
- Tex Wiki : 各種パッケージの利用
 - <https://texwiki.texjp.org/?LaTeX%E5%85%A5%E9%96%80%2F%E5%90%84%E7%A8%AE%E3%83%91%E3%83%83%E3%82%B1%E3%83%BC%E3%82%B8%E3%81%AE%E5%88%A9%E7%94%A8>

\backslash newcommand{}[]{}

- 独自のマクロを定義するコマンド
 - 定義したマクロが現れている箇所は、定義内容で置き換えられる
- 引数なしの命令の定義は「 \backslash newcommand{ \backslash 命令の名前} {定義内容}」
例) \backslash newcommand{ \backslash diff}{ d }
- 引数付きの命令の定義は「 \backslash newcommand{ \backslash 命令の名前}[引数の個数]{定義内容}」
例) \backslash newcommand{ \backslash combination}[2]{{}_{\#1} \mathrm{C}_{\#2}}
- TeX Wiki : LaTeXマクロの作成
 - <https://texwiki.texjp.org/?LaTeX%E5%85%A5%E9%96%80%2FLaTeX%E3%83%9E%E3%82%AF%E3%83%AD%E3%81%AE%E4%BD%9C%E6%88%90>

¥title{} と ¥author{} と ¥date{}

- 文書のタイトル、著者、日付を書く
- 論文を投稿する時は、指定されたテンプレートの類似の箇所に指定通り書けばよい
 - ¥date{}を使うことはあまりないでしょう。
- 本文中の¥maketitleの位置に、ここで書いた情報が現れる

¥begin{document}から¥end{document}

- この範囲が本文
- この範囲に書いたテキストがPDFに現れるが・・・
 - 改行を入れても改行されない
 - 改行を入れたいところで¥¥と書く
 - 空白行を入れると次の段落に移るだけ
 - 最初の段落を除いて、段落の先頭にはインデントが入る
 - 空白行を入れなくても¥parと書けばそこで次の段落に移る
 - 段落先頭のインデントを無くするには、段落の先頭に¥noindentと書く
- 体裁をコントロールするために様々なコマンドを使う

`¥begin{abstract}`から`¥end{abstract}`

- ここに要旨を書く
- 要旨がどのような体裁になるかは、`¥documentclass`コマンドで指定したクラスファイルの中で定義されている

章立てコマンド

- 本文を章、節などに分けるには、以下のコマンドを使う
 - 波括弧内に章、節などのタイトルを書く

<code>¥part{}</code>	部	レベル - 1
<code>¥chapter{}</code>	章	レベル 0
<code>¥section{}</code>	節	レベル 1
<code>¥subsection{}</code>	小節	レベル 2
<code>¥subsubsection{}</code>	小々節	レベル 3
<code>¥paragraph{}</code>	段落	レベル 4
<code>¥subparagraph{}</code>	小段落	レベル 5

演習3-1

- 白紙の文書に、以下の5つのセクションを追加してみよう
 - タイトルはMachine Learning、著者は自分の名前
 - 文章は無しでいいです
1. Introduction
 2. Related Work
 3. Proposed Method
 4. Empirical Evaluation
 5. Conclusions

文字の大きさ

- 「`{¥small}` ここは小さい文字」のように、サイズを変えたい部分を波括弧でくくり、先頭にコマンドを書く
 - 波括弧が重なっているときは、内側が優先
- サイズ変更コマンドは以下のとおり
 - https://www.overleaf.com/learn/latex/Font_sizes_families_and_styles

<code>¥tiny</code>
<code>¥scriptsize</code>
<code>¥footnotesize</code>
<code>¥small</code>
<code>¥normalsize</code>
<code>¥large</code>
<code>¥Large</code>
<code>¥LARGE</code>
<code>¥huge</code>
<code>¥Huge</code>

書体

- 書体は 3 通りの方法で分類される
 - ファミリー（文字のデザイン）、シリーズ（文字の太さ）、シェイプ（文字の見た目の形）
- 書体は宣言型かコマンド型で指定する
 - 宣言型は「{¥コマンド テキスト}」、コマンド型は「¥コマンド{テキスト}」
 - 宣言型を波括弧なしで使うと、以降のテキストすべてに影響を及ぼす

ファミリー	ローマン体	{¥rmfamily }	¥textrm{ }
	サンセリフ体	{¥sffamily }	¥textsf{ }
	タイプライタ体	{¥ttfamily }	¥texttt{ }
シリーズ	ミディアム	{¥mdseries }	
	ボールド	{¥bfseries }	¥textbf{ }
シェイプ	通常体	{¥upshape }	¥textup{ }
	イタリック体	{¥itshape }	¥textit{ }
	スモールキャップ体	{¥scshape }	¥textsc{ }
	斜体	{¥slshape }	¥textsl{ }

下線と強調

- 下線は`¥underline{}`
- 強調は`¥emph{}`
 - 強調は、単に斜体にすることではない
- 問：`¥textit{}`の中で`¥emph{}`を使ってみよう
- 問：`¥textbf{}`の中で`¥emph{}`を使ってみよう

リスト（箇条書き）

- 番号なしリストは`itemize`環境
- 番号付きリストは`enumerate`環境
- ラベル付きリストは`description`環境
- 使い方は同じ
 - `itemize`とある部分を`enumerate`や`description`に置き換える
 - ただし`description`環境では、例えば「`¥item[Point A] ...`」とラベルを角括弧内に書く

```
¥begin{itemize}  
  ¥item ...  
  ¥item ...  
  ¥item ...  
¥end{itemize}
```

演習3-2

- Introductionセクションに、以下の論文のIntroductionセクションの末尾にある箇条書きを、そのまま写経してみよう

<https://arxiv.org/abs/2003.06060>

- 最近の論文では、イントロの末尾に箇条書きで論文の貢献を列挙することが多いようです

We highlight our main contributions as follows:

- We provide more evidence that it is beneficial to sample from the energy-based model defined both by the generator and the discriminator instead of from the generator only.
- We derive an equivalent formulation of the pixel-space energy-based model in the latent space, and show that sampling is much more efficient in the latent space.
- We show experimentally that samples from this energy-based model are of higher quality than samples from the generator alone.
- We show that our method can approximately extend to other GAN formulations, such as Wasserstein GANs.

数式

LaTeXの本領発揮！

インライン・モードとディスプレイ・モード

- インライン・モードでは、数式を二つの\$記号で囲み、本文中に埋め込む

... the equation $E=mc^2$, discovered in 1905 by Albert Einstein ...

- ディスプレイ・モードでは、`\begin{}`～`\end{}`で囲まれた中に、数式を書く
 - `align`環境がおすすめ（ただし、`\usepackage{amsmath}`とプリアンブルに書いておく）
 - 式番号が必要なら、`align`の後の「*」を削除
 - `array`環境よりは良いはず

```
\begin{align*}
2x - 5y &= 8 \\
3x + 9y &= -12 \\
\end{align*}
```

上付き文字と下付き文字

- 上付き文字は「^」、下付き文字は「_」を、それぞれ使って書く

例) $c^{\{2\}}$

例) $a_{\{i\}}$

- https://www.overleaf.com/learn/latex/Subscripts_and_superscripts

括弧

- LaTeXでは様々な括弧を数式で使える
 - 波括弧は「 $\{$ 」 「 $\}$ 」
 - 二重の縦線の括弧は「 $|$ 」
 - 山括弧(angle brackets)は「 \angle 」 「 \rangle 」
- **amsmath**パッケージには様々な種類の括弧が用意されている
- 括弧の大きさは、例えば「 $($ 」であれば、以下のようにコントロールできる

$\big($ $\Big($ $\bigg($ $\Bigg($

分数

- 分数は、`\frac{分子}{分母}`と書く
- インライン・モードでもディスプレイ・モードでも同じ
- インライン・モードでは縦方向に詰まった感じになる
- ディスプレイ・モードでも、次のようにすると、縦方向に詰まった分数を使える

```
\begin{align*}f(x)=\frac{P(x)}{Q(x)} \quad \& \quad \& \text{term}{and} \quad \& \quad f(x)=\textstyle\frac{P(x)}{Q(x)} \\ \& \end{align*}
```

- 「`\&`」は少し空白を空けるという意味
- `\textrm{}`のように、`\text{??{}}`系の書体指定を使えば数式の中にテキストを書ける

数式を揃える

- 数式で等号や不等号をそろえたい場合は「&」を使う

```
\begin{align} \label{eq1} \\ \begin{split} A &= \frac{\pi r^2}{2} \\ &= \frac{1}{2} \pi r^2 \end{split} \\ \end{split} \\ \end{align}
```

- 複数の行に「&」を書くと、その位置で行が揃えられる
- `\label{eq1}`は数式へのラベルの指定
 - 本文中で、`\ref{eq1}`または`\eqref{eq1}`と書くと、その場所に数式番号が自動的に入る
- `split`環境を使わないと2つの行に別々に数式番号が打たれる

数式に関するその他諸々…

- 必要に応じてそのつと調べることが多いです
- Overleaf documentationのMathematicsの項にあるWebページ群を見てください

演習3-3

- 先ほどの論文の式(1)をSec. 3に追加してみよう
 - <https://arxiv.org/abs/2003.06060>

$$\begin{aligned} L_D &= -\mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] - \mathbb{E}_{z \sim p_z} [1 - \log D(G(z))] \\ L_G &= -\mathbb{E}_{z \sim p_z} [\log D(G(z))] \end{aligned} \tag{1}$$

図と表

`¥includegraphics[]{}`

- このコマンドを書いている場所の付近に波括弧内で指定した画像ファイルが挿入される
- プリアンブルで`¥graphicspath{}`コマンドを使い、画像のあるディレクトリを指定可
 - 書き方は`¥graphicspath{{dir1/}}{dir2/}{dir3/}...`

```
¥documentclass{article}
¥usepackage{graphicx}
¥graphicspath{ {./images/} }

¥begin{document}
The universe is immense and it seems to be homogeneous,
in a large scale, everywhere we look at.

¥includegraphics{universe}

There's a picture of a galaxy above
¥end{document}
```

¥includegraphics[]{}による画像の見た目の指定

- 角括弧の中に様々なオプションを指定できる
- スケールを指定 `¥includegraphics[scale=1.5]{lion-logo}`
- 実寸を指定 `¥includegraphics[width=3cm, height=4cm]{lion-logo}`
- 既定の長さを流用 `¥includegraphics[width=0.7¥textwidth]{universe}`
 - 参考： https://en.wikibooks.org/wiki/LaTeX/Page_Layout
- 反時計回りに回転 `¥includegraphics[angle=45]{lion-logo}`
- しかし...¥includegraphics[]{}だけを使うと図の位置がうまくコントロールできない

LaTeXで使う単位や長さ

pt	A point, is the default length unit. About 0.3515mm
mm	a millimeter
cm	a centimeter
in	an inch
ex	the height of an x in the current font
em	the width of an m in the current font
<code>\columnsep</code>	distance between columns
<code>\columnwidth</code>	width of the column
<code>\linewidth</code>	width of the line in the current environment
<code>\paperwidth</code>	width of the page
<code>\paperheight</code>	height of the page
<code>\textwidth</code>	width of the text
<code>\textheight</code>	height of the text
<code>\unitlength</code>	units of length in the <i>picture</i> environment.

https://www.overleaf.com/learn/latex/Inserting_Images#Reference_guide

`\begin{figure}[]\sim\end{figure}`

- `figure`環境を使うと図の位置をコントロールしやすい
- 角括弧の中に以下のようなパラメータ（次スライドも参照）を指定する
 - `h` おおよそここに図を配置
 - `t` ページの上端に図を配置
 - `b` ページの下端に図を配置
- `figure`環境の中で`\centering`コマンドを使うと図が中央揃えになる
- `figure`環境の中で`\caption{}`コマンドを使うとキャプションが付く
 - キャプションのテキストは波括弧の中に書く
- `figure`環境の中で`\label{}`コマンドを使うと参照用のラベルを図に付けられる
 - 波括弧の中にラベルの文字列を書く

図や表の位置決めに使うパラメータ

h	Place the float <i>here</i> , i.e., <i>approximately</i> at the same point it occurs in the source text (however, not <i>exactly</i> at the spot)
t	Position at the <i>top</i> of the page.
b	Position at the <i>bottom</i> of the page.
p	Put on a special <i>page</i> for floats only.
!	Override internal parameters LaTeX uses for determining "good" float positions.
H	Places the float at precisely the location in the LATEX code. Requires the float package, though may cause problems occasionally. This is somewhat equivalent to h! .

`\begin{figure}[]\sim\end{figure}`の例

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.25\textwidth]{mesh}
  \caption{a nice plot}
  \label{fig:mesh1}
\end{figure}
```

As you can see in the figure `\ref{fig:mesh1}`, the function grows near 0. Also, in the page `\pageref{fig:mesh1}` is the same example.

- 図のラベルは`{fig:...}`のように図のラベルだと分かるようにしておくとい
- `\pageref{ラベル}`は参照された図があるページのページ番号で置き換わる

`\begin{tabular}{} ~ \end{tabular}`

- 表を作るときは`tabular`環境を使う
- 表を中央揃えにするには`tabular`環境全体を`\begin{center} ~ \end{center}`で囲む
- 表の各列でテキストをどう配置するかは波括弧内のパラメータで指定する
 - `c`は中央揃え、`l`は左寄せ、`r`は右寄せ
 - `|`は罫線（`||`は二重線の罫線）
 - `\hline`は水平方向の罫線

```
\begin{center}
\begin{tabular}{|c||c|c| }
\hline
cell1 & cell2 & cell3 \&\& \hline\hline
cell4 & cell5 & cell6 \&\&
cell7 & cell8 & cell9 \&\&
\hline
\end{tabular}
\end{center}
```

`\begin{table}[] \sim \end{table}[]`

- 表にキャプションを付けるときは`tabular`環境全体を`table`環境で囲む
 - `table`環境の中で`\centering`コマンドを使うと表全体が中央揃えになる
 - 角括弧の中の書き方や`\caption{}`や`\label{}`の使い方などは`figure`環境と同じ

The table `\ref{table:1}` is an example of referenced `\LaTeX` elements.

```
\begin{table}[h!]  
\centering  
\caption{Table to test captions and labels}  
\begin{tabular}{||r r c c||}  
  \hline  
  Col1 & Col2 & Col2 & Col3 ¥¥ [0.5ex]  
  \hline¥hline  
  1 & 6 & 87837 & 787 ¥¥  
  2 & 7 & 78 & 5415 ¥¥  
  3 & 545 & 778 & 7507 ¥¥  
  4 & 545 & 18744 & 7560 ¥¥  
  5 & 88 & 788 & 6344 ¥¥ [1ex]  
  \hline  
\end{tabular}  
\label{table:1}  
\end{table}
```

¥multicolumn{}{}{}

- 列を結合するときに使うコマンド
 - 一つ目の波括弧：結合したい列の数
 - 二つ目の波括弧：結合後の列内でのテキストの配置の仕方と縦の罫線の有無
 - 三つ目の波括弧：結合後の列内に配置するテキスト

```
¥begin{tabular}{|p{3cm}||p{3cm}|p{3cm}|p{3cm}| }
¥hline
¥multicolumn{4}{|c|}{Country List} ¥¥
¥hline
Country Name      or Area Name& ISO ALPHA 2 Code &ISO ALPHA 3 Code&ISO numeric Code¥¥
¥hline
Afghanistan      & AF      &AFG&    004¥¥
Aland Islands&    AX    &ALA    &248¥¥
Albania &AL &ALB&    008¥¥
Algeria      &DZ &DZA&    012¥¥
American Samoa&    AS    &ASM&016¥¥
Andorra& AD    &AND    &020¥¥
Angola& AO    &AGO&024¥¥
¥hline
¥end{tabular}
```

参考文献リスト

biblatexパッケージ

- 参考文献リストは**biblatex**パッケージを使って作成するとよい
- 拡張子が**.bib**のテキストファイルにすべての参考文献の情報を書いておく
 - 著者名、タイトル、雑誌名（or プロシーディングス名）、ページ数など
- `\printbibliography` コマンドを書いた位置に参考文献リストが作成される

```
\documentclass{article}

\usepackage{biblatex}
\addbibresource{sample.bib}

\begin{document}
Let's cite! The Einstein's journal paper \cite{einstein} and the Dirac's
book \cite{dirac} are physics related items.

\printbibliography

\end{document}
```

bibファイルを使うメリット

- 毎回書誌情報を入力しなくてすむ
- ACM DLやDBLPなどの論文データベースがbibtexデータを用意してくれている
 - これを自分のbibファイルにコピペするだけでよい

```
@article{DBLP:journals/corr/GoodfellowPMXWOCB14,  
  author    = {Ian J. Goodfellow and  
               Jean Pouget{-}Abadie and  
               Mehdi Mirza and  
               Bing Xu and  
               David Warde{-}Farley and  
               Sherjil Ozair and  
               Aaron C. Courville and  
               Yoshua Bengio},  
  title     = {Generative Adversarial Networks},  
  journal   = {CoRR},  
  volume    = {abs/1406.2661},  
  year      = {2014},  
  url       = {http://arxiv.org/abs/1406.2661},  
  archivePrefix = {arXiv},  
  eprint    = {1406.2661}  
}
```

DBLPのbibtexファイルは、
あまりキレイでないこともあるので
自分で手直したほうがいいかも…。

bib ファイルのエントリの種類

article	book	mvbook
inbook	bookinbook	suppbook
booklet	collection	mvcollection
incollection	suppcollection	manual
misc	online	patent
periodical	suppperiodical	proceedings
mvproceedings	inproceedings	reference
mvreference	inreference	report
set	thesis	unpublished
custom	conference	electronic
masterthesis	phdthesis	techreport

bibファイルのエントリ内のフィールド

abstract	addendum	afterword	annotate
author	authortype	bookauthor	bookpagination
booksubtitle	booktitle	chapter	commentator
date	doi	edition	editor
editortype	eid	entrysubtype	eprint
eprinttype	eprintclass	eventdate	eventtitle
file	foreword	holder	howpublished
indextitle	institution	introduction	isan
isbn	ismn	isrn	issue
issuesubtitle	issuetitle	iswc	journalsubtitle
journaltitle	label	language	library
location	mainsubtitle	maintitle	month
note	number	organization	origdate
origlanguage	origlocation	origpublisher	origtitle
pages	pagetotal	pagination	part
publisher	pubstate	reprinttitle	series
shortauthor	shortedition	shorthand	shorthandintro
shortjournal	shortseries	shorttitle	subtitle
title	translator	type	url
venue	version	volume	year

biblatexの詳細

- https://www.overleaf.com/learn/latex/Articles/Getting_started_with_BibLaTeX
 - ここでは詳細は割愛します。

課題3

- GANの原論文をarXivからダウンロードする
 - <https://arxiv.org/abs/1406.2661>
- 論文のSection 4の4.1までと、Section 5とを、そのまま"写経"する
 - 自分が論文を書くときに、他人の論文をこのように丸写しするのは、当然ダメです！
 - 従来研究の紹介は、元の論文の英文を必ずパラフレイズすること！
 - 自分の過去の研究の紹介であっても必ずパラフレイズすること！
 - Webのパラフレイズサービスは当てにならないので使わないこと！
 - 今回はあくまでもLaTeXの練習なので、丸写しします
 - その範囲に出てくる参考文献も含めて丸写ししてください
 - 文献番号は違っていてもよいです
- ソースをzipでかためたものと、PDFとを、提出してください

課題3のヒント

- <https://www.overleaf.com/learn/latex/Algorithms>

次回のための予習

- さらのSSDにUbuntuをインストールし、CUDAをインストールしていただきます。
- そして、PyTorchをインストールして、適当なサンプルプログラムをGPU上で動かしてみます。
 - 例えば https://imaging-solution.net/deep-learning/pytorch/pytorch_mnist_sample_program/
- Linuxコマンドを予習しておいてください。

次の次の回のための予習

- Google Colab の使い方を把握しておいてください

<https://www.youtube.com/watch?v=KCCzo31Oo8U>