**Evaluating LLM Knowledge Distillation on Quantized Models**

Intro

~~As Artificial Intelligence becomes more and more prevalent in the modern world(a), more and more corporations will not only start looking into using AI within their own work scope but also inevitably cutting costs with AI.This matters because the more precise AI models, like GPT, use more resources and parameters which directly means more cost. In fact the largrest model of GPT-2 used 1.5 billion parameters[0.5] and is left far behind GPT-3 which used 175 billion parameters[0]. However, AI is still in its early stages and there could be ways to optimize it further, cutting down costs and enhancing its precision [d]. As previously stated, there is a massive leap in terms of efficiency between GPT-2 and GPT-3 but there is only about a 1-year gap between the two of them.The main point here is that AI has yet to hit a roadblock and it is still an ever-growing field of study.~~

Our experiment’s goal is to provide a way to optimize various AI models by using quantization and knowledge distillation. Quantization and knowledge distillation have been proven to be effective ways to optimize an AI Model’s Effiency[1][2] with research results coming from UC Berkely stating that in practice it is possible to improve Neural Network Models (NNs) by “reduce the memory footprint and latency by a factor of 16x” [1]. Furthermore, with knowledge distillation, various neural network students have shown to be competitive or better than their teachers when it came to accuracy with the students requiring much less resources than their teacher model[3]. With all this in mind, we intend to take an already established AI model and use quantization on it, making a different version of it that is smaller and uses less resources. Then we will use knowledge distillation in order to teach it and refine its precision with the teacher model being the original AI Model. The model we are using is Llama 3.1-8B-Instruct and our ultimate goal with this experiment is to show an increase in performance of the quantized student model after it is fine-tuned with knowledge distillation.

Key Contributions/Ideas

There have been many evaluations on the different methods of compressing LLMs. Knowledge distillation became popular after a 2015 paper[4] by a couple of researchers at Google. The paper detailed a way to train models using knowledge distillation in a neural network. Quantization and pruning are also common methods of compression[5]. The combination of quantization and pruning has been shown to produce diminishing returns, and much less potential for gains than the combination of knowledge distillation and quantization [6].

Quantization suffers loss in performance from token flips, where a different token is output due to ambiguous token distribution and loss of precision from quantization[7]. We aim to explore whether knowledge distillation from a base model is able to reduce the losses in performance in its quantized counterpart.(add about gaps in the knowledge)

At its core, quantization is a way to optimize a program’s size and how many resources it requires for use[8]. A research paper written by students at University of California: Berkeley in June of 2021 detailed various methods of Quantization and how they can be used for an efficient Neural Network(NN) interface. They actually go over various methods of optimizing models like modifying the NN program itself, creating hardware purpose made for NNs, pruning the program and more. However, they leave off at Quantization itself, saying that it has shown “Great and consistent success in both training and interference of NN Models.”. Quantization improves an AI model’s speed and performance by decreasing the amount of parameters the model itself has. This is usually done via changing the weights like a 32-bit precision to a lower precision like 16 bit or 8 bit, etc[9]. The most pressing concern about quantization would be the gap in quality and how a quantized neural network (QNN) would differ from the base model. In truth, the gap is negligible according to a research article in written in 2024 called “4.6-Bit Quantization for Fast and Accurate Neural Network Inference on CPUs”[9]. Researchers tested 4.6 bit quantization and discovered that the quality is close to the mean of the 4-bit and 8-bit neural networks while being 1.5-1.6 times faster than the 8-bit neural network. In this context, each model was of the same architecture. Quantization can improve a model’s performance while at the same time decreasing the cost of the model and we believe that it is possible to improve upon that.

Knowledge distillation is the process of training a large scale AI model training a smaller model with the ultimate goal of the smaller model being able to solve the same task as the large scale one with an improved performance. In this context, the large scale AI model is referred to as the teacher and the smaller model is referred to as the student[10]. Knowledge distillation can be broken down into three separate parts: the knowledge itself or what you want the student to learn, the algorithm for the teacher model to teach the student model, and lastly the teacher-student architecture[10]. There are many pathways you can go about when it comes to knowledge distillation but for the purpose of our experiment, we are using something very similar to quantized distillation. In quantized distillation, after the teacher model has already been fed the information, they then proceed to teach it to a quantized version of themselves and use that model to teach an apprentice model with the similar or the same parameters[10]. In our process though, we are not using another model and want to focus on what happens when a teacher model teaches a quantized student model of itself.

Methods

~~We will evaluate LLama 3.1-8B on [n] benchmarks and 4-bit and 8-bit quantized LLama 3.1-8B on the same benchmarks. We will use Llama 3.1-8B as the teacher, and evaluate how knowledge distillation affects the performance of the quantized Llama models.~~

As stated previously, our experiment’s goal is to study the process of quantizing an AI model then train the model using Knowledge Distillation, with the student model being the quantized AI model and the teacher model the original. The AI we chose for this was LLama 3.1-8B and we want the quantized version of it to be able to achieve an accuracy score greater than the teacher’s on the selected data set. The main reason we chose llama was due to resource constraints. We spent a lot of time thinking about the various AI models we could use for this and we narrowed it down to GPT3 6.7B and LLama 3.1-8B. We went with Llama, mainly because it was designed to be cost-effective and was available in the huggingface. We did not have a large amount of funds and so cost was a concern with this experiment.

The knowledge distillation pipeline was also chosen with cost in mind. We chose to use Logit-Based knowledge Distillation, not only for its cost efficiency, but also it doesn’t require us to access some of the deeper parts of the language model [12]. LLama may be open source, but Logits-Based KD also allows us to use less optimized equipment which meant the entire experiment could be run in google colaboratory. The main issue with logit distillation is that there is a performance gap when it comes down to other forms of KDs but recent advances in Logit-Based KD have been narrowing that performance gap.

The dataset we are training the model on is GSM8K but other researchers may be inclined to try out other datasets to see if there is a difference with their own experiment and this one. For us, we are comparing 4-bit with 16-bit models and as stated prior, planned to use Logit Based knowledge distillation to improve the performance of the student model. Before we applied the actual Knowledge Distillation, the 4-bit Llama student model had a reported accuracy score on GSM8K at 76.2%. That makes the bare minimum end goal be greater than 76.2% but ideally we would want the student model to be greater than 84.5%, which is the current accuracy score for 16 bit Llama 3.1-8B tested on GSM8K[11]. The teacher model will be an unquantized version of Llama 3.1-8B which is 16 bit. We will get the logits on GSM8K then use those logits to train the 4-bit model. If everything goes well, the 4-bit trained model should achieve a higher accuracy score than its untrained self and that score should at least be competitive when compared to the teacher model.

Experimental Setup

Quantize the student 1 model do

The student model was quantized in google colab.

Test student-1’s performance

Outline:

* Step 1:
* Quantization of Model(Source code + screenshot )(I need someone to
* Benchmarks of AI Model when tested on
* Step 2:
* Knowledge Distillation process()
* Source Code
* Step 3:
* Run Benchmarks again and compare results

Datasets

1. GSM8K

Benchmarks

1. GSM8K

Compute requirements

* Quantize the student 1 model
* Training student models
* Testing the teacher model
* Testing student models

Ideal results

A clear difference between the performance of the quantized student models before and after knowledge distillation. At the start, the models originally

Potential Limitations

**References**

[1]<https://openaccess.thecvf.com/content_ICCV_2019/papers/Cho_On_the_Efficacy_of_Knowledge_Distillation_ICCV_2019_paper.pdf>

[2] <https://arxiv.org/abs/2103.13630> ]

1 <https://openaccess.thecvf.com/content_ICCV_2019/papers/Cho_On_the_Efficacy_of_Knowledge_Distillation_ICCV_2019_paper.pdf>

[4] Distilling the Knowledge in a Neural Network<https://arxiv.org/pdf/1503.02531>

[5] Compressing LLMs: The Truth is Rarely Pure and Never Simple

<https://www.semanticscholar.org/paper/Compressing-LLMs%3A-The-Truth-is-Rarely-Pure-and-Jaiswal-Gan/4e13ecf80443a4135d516b7ba77eca82b5c6d347>

[6] Combining Compressions for Multiplicative Size Scaling on Natural Language Tasks <https://arxiv.org/pdf/2208.09684>

[7] Improving Conversational Abilities of Quantized Large Language Models via Direct Preference Alignment <https://arxiv.org/pdf/2407.03051>

[8]A Survey of Quantization Methods for Efficient Neural Network Inference

<https://arxiv.org/pdf/2103.13630>

[9]

<https://www.mdpi.com/2227-7390/12/5/651>

[10]

<https://link.springer.com/article/10.1007/s11263-021-01453-z>

[11]

<https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct>

[12] <https://openaccess.thecvf.com/content/CVPR2023/papers/Jin_Multi-Level_Logit_Distillation_CVPR_2023_paper.pdf>

[a] Artificial intelligence in modern society <https://digitalcommons.murraystate.edu/cgi/viewcontent.cgi?article=1148&context=bis437>

[b] AI-Based Modeling: Techniques Applications and Research Issues Towards Automation, Intelligent and Smart Systems <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8830986/>

[c] ​​Beyond implementation: the long-term economic impact of AI in healthcare <https://www.tandfonline.com/doi/full/10.1080/13696998.2023.2285186?scroll=top&needAccess=true#d1e109>

[d] Artificial intelligence: A powerful paradigm for scientific research <https://www.sciencedirect.com/science/article/pii/S2666675821001041>

LLM-QAT: Data-Free Quantization Aware Training for Large Language Models

<https://www.semanticscholar.org/reader/6bd3ee1ca608bc66a490f63f2fb107d79b44f3e2>

Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes

<https://arxiv.org/pdf/2305.02301v2>

Combining Compressions for Multiplicative Size Scaling on Natural Language Tasks

<https://arxiv.org/pdf/2208.09684>

ZeroQuant: Efficient and Affordable Post-Training Quantization for Large-Scale Transformers

<https://arxiv.org/pdf/2206.01861>

[6]

<https://www.mdpi.com/2227-7390/12/5/651>

[7]

<https://huggingface.co/docs/optimum/en/concept_guides/quantization>

[8]

<https://link.springer.com/article/10.1007/s11263-021-01453-z>

[9]

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8053015/>

[0] [**https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10166793/**](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10166793/)

**[0.5]** [**https://huggingface.co/transformers/v2.11.0/model\_doc/gpt2.html**](https://huggingface.co/transformers/v2.11.0/model_doc/gpt2.html)

% This must be in the first 5 lines to tell arXiv to use pdfLaTeX, which is strongly recommended.

\pdfoutput=1

% In particular, the hyperref package requires pdfLaTeX in order to break URLs across lines.

\documentclass[11pt]{article}

% Change "review" to "final" to generate the final (sometimes called camera-ready) version.

% Change to "preprint" to generate a non-anonymous version with page numbers.

\usepackage[review]{coling}

% Standard package includes

\usepackage{times}

\usepackage{latexsym}

% For proper rendering and hyphenation of words containing Latin characters (including in bib files)

\usepackage[T1]{fontenc}

% For Vietnamese characters

% \usepackage[T5]{fontenc}

% See https://www.latex-project.org/help/documentation/encguide.pdf for other character sets

% This assumes your files are encoded as UTF8

\usepackage[utf8]{inputenc}

% This is not strictly necessary, and may be commented out,

% but it will improve the layout of the manuscript,

% and will typically save some space.

\usepackage{microtype}

% This is also not strictly necessary, and may be commented out.

% However, it will improve the aesthetics of text in

% the typewriter font.

\usepackage{inconsolata}

%Including images in your LaTeX document requires adding

%additional package(s)

\usepackage{graphicx}

% If the title and author information does not fit in the area allocated, uncomment the following

%

%\setlength\titlebox{<dim>}

%

% and set <dim> to something 5cm or larger.

\title{Instructions for COLING 2025 Proceedings}

% Author information can be set in various styles:

% For several authors from the same institution:

% \author{Author 1 \and ... \and Author n \\

% Address line \\ ... \\ Address line}

% if the names do not fit well on one line use

% Author 1 \\ {\bf Author 2} \\ ... \\ {\bf Author n} \\

% For authors from different institutions:

% \author{Author 1 \\ Address line \\ ... \\ Address line

% \And ... \And

% Author n \\ Address line \\ ... \\ Address line}

% To start a separate ``row'' of authors use \AND, as in

% \author{Author 1 \\ Address line \\ ... \\ Address line

% \AND

% Author 2 \\ Address line \\ ... \\ Address line \And

% Author 3 \\ Address line \\ ... \\ Address line}

\author{First Author \\

Affiliation / Address line 1 \\

Affiliation / Address line 2 \\

Affiliation / Address line 3 \\

\texttt{email@domain} \\\And

Second Author \\

Affiliation / Address line 1 \\

Affiliation / Address line 2 \\

Affiliation / Address line 3 \\

\texttt{email@domain} \\}

%\author{

% \textbf{First Author\textsuperscript{1}},

% \textbf{Second Author\textsuperscript{1,2}},

% \textbf{Third T. Author\textsuperscript{1}},

% \textbf{Fourth Author\textsuperscript{1}},

%\\

% \textbf{Fifth Author\textsuperscript{1,2}},

% \textbf{Sixth Author\textsuperscript{1}},

% \textbf{Seventh Author\textsuperscript{1}},

% \textbf{Eighth Author \textsuperscript{1,2,3,4}},

%\\

% \textbf{Ninth Author\textsuperscript{1}},

% \textbf{Tenth Author\textsuperscript{1}},

% \textbf{Eleventh E. Author\textsuperscript{1,2,3,4,5}},

% \textbf{Twelfth Author\textsuperscript{1}},

%\\

% \textbf{Thirteenth Author\textsuperscript{3}},

% \textbf{Fourteenth F. Author\textsuperscript{2,4}},

% \textbf{Fifteenth Author\textsuperscript{1}},

% \textbf{Sixteenth Author\textsuperscript{1}},

%\\

% \textbf{Seventeenth S. Author\textsuperscript{4,5}},

% \textbf{Eighteenth Author\textsuperscript{3,4}},

% \textbf{Nineteenth N. Author\textsuperscript{2,5}},

% \textbf{Twentieth Author\textsuperscript{1}}

%\\

%\\

% \textsuperscript{1}Affiliation 1,

% \textsuperscript{2}Affiliation 2,

% \textsuperscript{3}Affiliation 3,

% \textsuperscript{4}Affiliation 4,

% \textsuperscript{5}Affiliation 5

%\\

% \small{

% \textbf{Correspondence:} \href{mailto:email@domain}{email@domain}

% }

%}

\begin{document}

\maketitle

\begin{abstract}

This document is a supplement to the general instructions for COLING 2025 authors. It contains instructions for using the \LaTeX{} style files for COLING 2025.

The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like.

These instructions should be used both for papers submitted for review and for final versions of accepted papers.

\end{abstract}

\section{Introduction}

Our experiment’s goal is to provide a way to optimize various AI models by using quantization and knowledge distillation. Quantization and knowledge distillation have been proven to be effective ways to optimize an AI Model’s Efficiency \cite{Cho:1} \cite{Kim:2} with research results coming from the University of California: Berkeley stating that in practice it is possible to improve Neural Network Models (NNs) by “reduce the memory footprint and latency by a factor of 16x” \cite{Kim:2}. Furthermore, with knowledge distillation, various neural network students have shown to be competitive or better than their teachers when it came to accuracy with the students requiring much less resources than their teacher model\cite{Cho:1}. With all this in mind, we intend to take an already established AI model and use quantization on it, making a different version of it that is smaller and uses less resources. Then we will use knowledge distillation in order to teach it and refine its precision with the teacher model being the original AI Model. The model we are using is Llama 3.1-8B-Instruct and our ultimate goal with this experiment is to show an increase in performance of the quantized student model after it is fine-tuned with knowledge distillation.

\section{Key Contributions/Ideas}

There have been many evaluations on the different methods of compressing LLMs. Knowledge distillation became popular after a 2015 paper\cite{Jaiswal:4} by a couple of researchers at Google. The paper detailed a way to train models using knowledge distillation in a neural network. Quantization and pruning are also common methods of compression\cite{Movva:5} but we decided to primarily focus on quantization mainly because (BENEFIT OF Q OVER P)

Quantization suffers loss in performance from token flips, where a different token is output due to ambiguous token distribution and loss of precision from quantization\cite{Lee:6}. We aim to explore whether or not knowledge distillation can be an effective way to improve performance of quantized model when you compare it's testing benchamrks to its original model.

At its core, quantization is a way to optimize a program’s size and how many resources it requires for use\cite{Kim:2}. A research paper written by students at University of California: Berkeley in June of 2021 detailed various methods of Quantization and how they can be used for an efficient Neural Network(NN) interface. They actually go over various methods of optimizing models like modifying the NN program itself, creating hardware purpose made for NNs, pruning the program and more. However, they leave off at Quantization itself, saying that it has shown “Great and consistent success in both training and interference of NN Models.”. Quantization improves an AI model’s speed and performance by decreasing the amount of parameters the model itself has. This is usually done via changing the weights like a 32-bit precision to a lower precision like 16 bit or 8 bit, etc\cite{Trusov:7}. The most pressing concern about quantization would be the gap in quality and how a quantized neural network (QNN) would differ from the base model. In truth, the gap is negligible according to a research article in written in 2024 called “4.6-Bit Quantization for Fast and Accurate Neural Network Inference on CPUs”\cite{Trusov:7}. Researchers tested 4.6 bit quantization and discovered that the quality is close to the mean of the 4-bit and 8-bit neural networks while being 1.5-1.6 times faster than the 8-bit neural network. In this context, each model was of the same architecture. Quantization can improve a model’s performance while at the same time decreasing the cost of the model and we believe that it is possible to improve upon that.

Knowledge distillation is the process of training a large scale AI model training a smaller model with the ultimate goal of the smaller model being able to solve the same task as the large scale one with an improved performance. In this context, the large scale AI model is referred to as the teacher and the smaller model is referred to as the student\cite{Gou:8}. Knowledge distillation can be broken down into three separate parts: the knowledge itself or what you want the student to learn, the algorithm for the teacher model to teach the student model, and lastly the teacher-student architecture\cite{Gou:8}. There are many pathways you can go about when it comes to knowledge distillation but for the purpose of our experiment, we are using something very similar to quantized distillation. In quantized distillation, after the teacher model has already been fed the information, they then proceed to teach it to a quantized version of themselves and use that model to teach an apprentice model with the similar or the same parameters\cite{Gou:8}. In our process though, we are not using another model and want to focus on what happens when a teacher model teaches a quantized student model of itself.

\section{Methodology}

As stated previously, our experiment’s goal is to study the process of quantizing an AI model then begin to train that model by using Knowledge Distillation, with the student model being the quantized AI model and the teacher model the original. The Language Model we chose for this was Llama 3.1-8B and we want the quantized trained version of it to have a accuracy score significantly higher than its untrained variant. The accuracy scores will be obtained using the GSM8K dataset. The main reason we chose Llama was due to resource constraints. We spent a lot of time thinking about the various AI models we could use for this and we narrowed it down to GPT3 6.7B and LLama 3.1-8B. We went with Llama, mainly because it was more cost effective compared to GPT\cite{COSTCOM:12}. We did not have a large amount of resources and so cost was a concern with this experiment.

The knowledge distillation pipeline was also chosen with cost in mind. We chose to use Logit-Based knowledge distillation, not only for its cost efficiency\cite{Park:13}, but also it doesn’t require us to access some of the deeper parts of the language model\cite{Tang:15}. Llama may be open source, but Logits-Based KD also allows us to use less optimized equipment\cite{Sun:14} which meant the entire experiment could be run in google colaboratory. The main issue with logit distillation is that there is a performance gap when it comes down to other forms of KDs\cite{Sun:14}, but the storage efficiency paired with the less strict computing requirements \cite{Lightly:16}was something we couldn't really reject.

The dataset we are training the model on is GSM8K but other researchers may be inclined to try out other datasets to see if there is a difference with their own experiment and this one. For us, we are comparing 4-bit with 16-bit models and as stated prior, planned to use Logit Based knowledge distillation to improve the performance of the student model. Before we applied the actual Knowledge Distillation, the 4-bit Llama student model had a reported accuracy score on GSM8K at 76.2 \% . That makes the bare minimum end goal be greater than 76.2\% but also, the most idealistic result we would like to get would to have the student model achieve an accuracy score greater than 84.5\%. 84.5 is the current accuracy score for 16 bit Llama 3.1-8B tested on GSM8K\cite{HFCO:11} The teacher model will be an unquantized version of Llama 3.1-8B which is 16 bit. We will get the logits on GSM8K then use those logits to train the 4-bit model. If everything goes well, the 4-bit trained model should achieve a decently higher accuracy score than its untrained self.

\section{Discussion}

\section{Preamble}

The first line of the file must be

\begin{quote}

\begin{verbatim}

\documentclass[11pt]{article}

\end{verbatim}

\end{quote}

To load the style file in the review version:

\begin{quote}

\begin{verbatim}

\usepackage[review]{coling}

\end{verbatim}

\end{quote}

For the final version, omit the \verb|review| option:

\begin{quote}

\begin{verbatim}

\usepackage{coling}

\end{verbatim}

\end{quote}

To use Times Roman, put the following in the preamble:

\begin{quote}

\begin{verbatim}

\usepackage{times}

\end{verbatim}

\end{quote}

(Alternatives like txfonts or newtx are also acceptable.)

Please see the \LaTeX{} source of this document for comments on other packages that may be useful.

Set the title and author using \verb|\title| and \verb|\author|. Within the author list, format multiple authors using \verb|\and| and \verb|\And| and \verb|\AND|; please see the \LaTeX{} source for examples.

By default, the box containing the title and author names is set to the minimum of 5 cm. If you need more space, include the following in the preamble:

\begin{quote}

\begin{verbatim}

\setlength\titlebox{<dim>}

\end{verbatim}

\end{quote}

where \verb|<dim>| is replaced with a length. Do not set this length smaller than 5 cm.

\section{Document Body}

\subsection{Footnotes}

Footnotes are inserted with the \verb|\footnote| command.\footnote{This is a footnote.}

\subsection{Tables and figures}

See Table~\ref{tab:accents} for an example of a table and its caption.

\textbf{Do not override the default caption sizes.}

\begin{table}

\centering

\begin{tabular}{lc}

\hline

\textbf{Command} & \textbf{Output} \\

\hline

\verb|{\"a}| & {\"a} \\

\verb|{\^e}| & {\^e} \\

\verb|{\`i}| & {\`i} \\

\verb|{\.I}| & {\.I} \\

\verb|{\o}| & {\o} \\

\verb|{\'u}| & {\'u} \\

\verb|{\aa}| & {\aa} \\\hline

\end{tabular}

\begin{tabular}{lc}

\hline

\textbf{Command} & \textbf{Output} \\

\hline

\verb|{\c c}| & {\c c} \\

\verb|{\u g}| & {\u g} \\

\verb|{\l}| & {\l} \\

\verb|{\~n}| & {\~n} \\

\verb|{\H o}| & {\H o} \\

\verb|{\v r}| & {\v r} \\

\verb|{\ss}| & {\ss} \\

\hline

\end{tabular}

\caption{Example commands for accented characters, to be used in, \emph{e.g.}, Bib\TeX{} entries.}

\label{tab:accents}

\end{table}

As much as possible, fonts in figures should conform

to the document fonts. See Figure~\ref{fig:experiments} for an example of a figure and its caption.

Using the \verb|graphicx| package graphics files can be included within figure

environment at an appropriate point within the text.

The \verb|graphicx| package supports various optional arguments to control the

appearance of the figure.

You must include it explicitly in the \LaTeX{} preamble (after the

\verb|\documentclass| declaration and before \verb|\begin{document}|) using

\verb|\usepackage{graphicx}|.

\begin{figure}[t]

\includegraphics[width=\columnwidth]{example-image-golden}

\caption{A figure with a caption that runs for more than one line.

Example image is usually available through the \texttt{mwe} package

without even mentioning it in the preamble.}

\label{fig:experiments}

\end{figure}

\begin{figure\*}[t]

\includegraphics[width=0.48\linewidth]{example-image-a} \hfill

\includegraphics[width=0.48\linewidth]{example-image-b}

\caption {A minimal working example to demonstrate how to place

two images side-by-side.}

\end{figure\*}

\subsection{Hyperlinks}

Users of older versions of \LaTeX{} may encounter the following error during compilation:

\begin{quote}

\verb|\pdfendlink| ended up in different nesting level than \verb|\pdfstartlink|.

\end{quote}

This happens when pdf\LaTeX{} is used and a citation splits across a page boundary. The best way to fix this is to upgrade \LaTeX{} to 2018-12-01 or later.

\subsection{Citations}

\begin{table\*}

\centering

\begin{tabular}{lll}

\hline

\textbf{Output} & \textbf{natbib command} & \textbf{ACL only command} \\

\hline

\citep{Gusfield:97} & \verb|\citep| & \\

\citealp{Gusfield:97} & \verb|\citealp| & \\

\citet{Gusfield:97} & \verb|\citet| & \\

\citeyearpar{Gusfield:97} & \verb|\citeyearpar| & \\

\citeposs{Gusfield:97} & & \verb|\citeposs| \\

\hline

\end{tabular}

\caption{\label{citation-guide}

Citation commands supported by the style file.

The style is based on the natbib package and supports all natbib citation commands.

It also supports commands defined in previous ACL style files for compatibility.

}

\end{table\*}

Table~\ref{citation-guide} shows the syntax supported by the style files.

We encourage you to use the natbib styles.

You can use the command \verb|\citet| (cite in text) to get ``author (year)'' citations, like this citation to a paper by \citet{Gusfield:97}.

You can use the command \verb|\citep| (cite in parentheses) to get ``(author, year)'' citations \citep{Gusfield:97}.

You can use the command \verb|\citealp| (alternative cite without parentheses) to get ``author, year'' citations, which is useful for using citations within parentheses (e.g. \citealp{Gusfield:97}).

A possessive citation can be made with the command \verb|\citeposs|.

This is not a standard natbib command, so it is generally not compatible

with other style files.

\subsection{References}

\nocite{Ando2005,andrew2007scalable,rasooli-tetrault-2015}

The \LaTeX{} and Bib\TeX{} style files provided roughly follow the American Psychological Association format.

If your own bib file is named \texttt{custom.bib}, then placing the following before any appendices in your \LaTeX{} file will generate the references section for you:

\begin{quote}

\begin{verbatim}

\bibliography{custom}

\end{verbatim}

\end{quote}

You can obtain the complete ACL Anthology as a Bib\TeX{} file from \url{https://aclweb.org/anthology/anthology.bib.gz}.

To include both the Anthology and your own .bib file, use the following instead of the above.

\begin{quote}

\begin{verbatim}

\bibliography{anthology,custom}

\end{verbatim}

\end{quote}

Please see Section~\ref{sec:bibtex} for information on preparing Bib\TeX{} files.

\subsection{Equations}

An example equation is shown below:

\begin{equation}

\label{eq:example}

A = \pi r^2

\end{equation}

Labels for equation numbers, sections, subsections, figures and tables

are all defined with the \verb|\label{label}| command and cross references

to them are made with the \verb|\ref{label}| command.

This an example cross-reference to Equation~\ref{eq:example}.

\subsection{Appendices}

Use \verb|\appendix| before any appendix section to switch the section numbering over to letters. See Appendix~\ref{sec:appendix} for an example.

\section{Bib\TeX{} Files}

\label{sec:bibtex}

Unicode cannot be used in Bib\TeX{} entries, and some ways of typing special characters can disrupt Bib\TeX's alphabetization. The recommended way of typing special characters is shown in Table~\ref{tab:accents}.

Please ensure that Bib\TeX{} records contain DOIs or URLs when possible, and for all the ACL materials that you reference.

Use the \verb|doi| field for DOIs and the \verb|url| field for URLs.

If a Bib\TeX{} entry has a URL or DOI field, the paper title in the references section will appear as a hyperlink to the paper, using the hyperref \LaTeX{} package.

\section\*{Acknowledgments}

This document has been adapted by Emily Allaway from the instructions for earlier ACL and NAACL proceedings, including those for NAACL 2024 by Steven Bethard, Ryan Cotterell and Rui Yan,

ACL 2019 by Douwe Kiela and Ivan Vuli\'{c},

NAACL 2019 by Stephanie Lukin and Alla Roskovskaya,

ACL 2018 by Shay Cohen, Kevin Gimpel, and Wei Lu,

NAACL 2018 by Margaret Mitchell and Stephanie Lukin,

Bib\TeX{} suggestions for (NA)ACL 2017/2018 from Jason Eisner,

ACL 2017 by Dan Gildea and Min-Yen Kan,

NAACL 2017 by Margaret Mitchell,

ACL 2012 by Maggie Li and Michael White,

ACL 2010 by Jing-Shin Chang and Philipp Koehn,

ACL 2008 by Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui,

ACL 2005 by Hwee Tou Ng and Kemal Oflazer,

ACL 2002 by Eugene Charniak and Dekang Lin,

and earlier ACL and EACL formats written by several people, including

John Chen, Henry S. Thompson and Donald Walker.

Additional elements were taken from the formatting instructions of the \emph{International Joint Conference on Artificial Intelligence} and the \emph{Conference on Computer Vision and Pattern Recognition}.

% Bibliography entries for the entire Anthology, followed by custom entries

%\bibliography{anthology,custom}

% Custom bibliography entries only

\bibliography{custom}

\appendix

\section{Example Appendix}

\label{sec:appendix}

This is an appendix.

\end{document}