Extending BERT for multi-choice problems

Orr Dermer [*1] **Afek Adler** [*2]

Abstract

Deep learning models are still far away from human-level performance in commonsense reasoning tasks. We introduce a multi-choice question answering scheme that models the dependencies between answers by building a relationship graph between the answers for a given question. then, we compute a centrality measure for each node (answer) in the graph and predict the answer by taking the node with the maximum value of that centrality measure. We test ourselves on the challenging CommonsenseQA dataset and show that although our method is capable of modeling more complex relationships between the answers we did not manage to improve the current state of the art on this task.

1. Introduction

When we face a multi-choice question answering (QA) problem where we have to choose the most probable answer we arrive at it with prior information and knowledge. for instance, we have some kind of a

Whereas in some cases the state of the art machine-learning methods which were trained on massive amount of data to infer prediction roles outperforms human-level by a large margin (e.g. SQUAD 2.0 (Rajpurkar et al., 2018)), In the field of common sense reasoning (Talmor et al., 2018) state of the art algorithms fail to generalize and are yet inferior to human level performance. this is due to two main reasons. first, is that when using algorithms such as fine-tuning a pre-trained language model, it is likely that this knowledge (common sense) does not exist explicitly in the pre training corpus. and second, is due to the fact that creating massive datasets for this purpose solely is hard, and developed only in the last years mostly by crowdsourcing.

[*]Equal contribution [1]Department of Computer Science, University of Tel Aviv, Tel Aviv, Israel. [2]Department of Industrial Engineering, University of Tel Aviv, Tel Aviv, Israel. Correspondence to: Afek Adler <afekilayadler@gmail.com>, Orr Dermer <odermer@gmail.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

Due to this inherent difficulty of this methods the research community invests resources in creating larger-scale datasets in different difficulty levels, e.g., the SWAG (Zellers et al., 2018) dataset with 100K examples where (Devlin et al., 2018) demonstrate a model which is already at a human level and the CommonseneQA (Talmor et al., 2018) a dataset which was made especially for common sense reasoning with ˜10K samples where the state of the art is much lower then human-level. for a detailed survey of resources see (Storks et al., 2019).

In this work, we suggest and evaluate models which eliminate the Independence assumption between answers for a given question, aiming to improve the benchmark on the CommonsenseQA dataset.

2. Background and Related Work

Research on machine common sense QA has long been acknowledged for it's critical component in natural language understanding. but only recently with the embracing of deep learning in natural language processing and crowdsourcing to create larger scale datasets, major breakthroughs has emerged; to the best of the author's knowledge, the first large scale common sense reasoning QA dataset was introduced is 2018 (Storks et al., 2019).

Due to the fact that the prominent approach to to solve common sense QA problems is with word embeddings (WE) and contextualized word embeddings (CWE) in the next section we will provide a brief introduction to CVE and explain how to use it in the context of common sense QA.

2.1. Contextualized Word Embeddings

Based on the pioneering work of Word2Vec (Mikolov et al., 2013) which enabled words to be represented as vectors, those enriching the information contained in the input for a variation of language based machine learning tasks, many models extended the Word2Vec concept by taking context into account; the same word has a different meaning in different context. the leading approaches to obtain a contextualized word representations is using the hidden layer of an LSTM (Hochreiter & Schmidhuber, 1997) and more specifically a biLSTM such as in ELMO (Peters et al., 2018) or more recently using a transformer architecture such as

(Devlin et al., 2018; Yang et al., 2019).

The great power of WE or CWE is to enable using a generic method as a basis for task specific machine learning models; for instance, in natural language processing there are many different tasks (e.g. machine translation, sentiment analysis) and a simple model based on CWE with fine tuning on a given dataset usually provides much better results than previous models on a given task, creating a new bencmark for this specific task. the fact that mega companies provides those trained models as an open source service makes it easy to use for academics and in the industry as one.

CWE is a type of transfer learning method; a model is pre-trained on a base dataset and fine tuned on the task dataset, with the underlying hypothesis that information distilation is possible from one source (base dataset) to the other (task specific dataset). The main reason for the great success of CWE is that the base model is trained using supervised learning on huge amounts of data, e.g. BERT, was trained on 3.3B words. the fact that CWE models are also self trained makes it easier to collect such huge amount of data without the need to label examples. the downside of this models is that they use many parameters (BERT Large - 340M) - so trainig and infering requires large processing power and latency is a challenge for creating real time systems. in the next section we will introduce the benchmark model which we aim to improve - BERT for CommonsenseQA (which is identical to BERT for SWAG) as was introduced in the original paper of BERT.

2.2. BERT

BERT is a word based CWE model that was pre-trained on two learning objectives. given a sentence, predict masked words (15% of the words were masked) and next sentence prediction. BERTs model architecture is a multi-layer bidirectional Transformer encoder (Vaswani et al., 2017) with the following hyper-parameters (BERT Large) - 12 Transformer blocks, hidden size 1024 (H) and 16 self attention heads. BERT input can be one or two sentences.

BERT input representation (figure 1) uses WordPeace (Wu et al., 2016) tokenization which is a data driven method that aims to achieve a balance between vocabulary size and out-of-vocab words. this way BERT stores only 30K vocabulary words very rarely encounter out-of-vocab words when tokenizing English texts. two special token are added for each sentence, a start token [CLS] and an end token [SEP]. if the input contains two sentences another [CLS] token is added in between the sentences. Due to the fact that Transformers do not encode the sequential nature of their inputs a segment ids and position embeddings are added as well to the input.

fine tuning BERT is simply swapping out the appropriate input and output, whether the input is a single text or a text pair the input and output pairs are given to the model and the parameters are fine-tuned by back-propagation end to end after connecting to the [CLS] token embedding to a task specific classification layer with a given loss function.

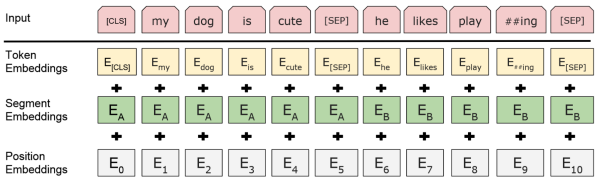

Figure 1. input representation for BERT (from the original paper).

2.3. BERT baseline for SWAG

In the BERT Paper, the authors approach the SWAG challange with the following input (input 1). that input is fed to BERT architecture and the [CLS] token embedding is used to predict a logit by calculating a dot product with a learned vector. this process is done for each answer independently of each other and then a softmax layer with cross entropy loss is calculated to compute the error. As we can see, this architecture introduce only H (1024) new parameters to the existing BERT model.

$$[CLS][Question][SEP][Answer][SEP] \quad (1)$$

3. CommonSeneseQA

CommonSeneseQA is a dataset which specifically made for common sense reasoning, by asking crowd workers to generate questions from concepts from ConceptNet. it contains 12,247 examples and 5 answers per question. the Authors use the same architecture as in the BERT benchmark for SWAG and report Accuracy score of 56.7%. The current state of the art on this dataset is currently 68.3% accuracy, 23.6% below human level.

4. Methodology

We propose a method with different variants to model the dependencies between answers, by feeding answer pairs jointly to BERT's architecture. by doing so, we contribute by:

- making the model more expressive by removing the independence assumption
- extending the model to deal with questions as 'what is largest?"

- extending the model to deal with answers as 'all answer are correct" or 'none of the above"

We derive our model from the BERT fine-tuning method over multichoice questions, as displayed in the original paper. However, as we want our model to compare between the different possible answers, we deviate from the original model by introducing the model with the question and 2 possible answers as each input sequence, for which the model returns a score - 1 meaning the first answer is more likely, and 0 if the second answer is the likeliest. Having received a score for each of pairs, we then integrate those into probability scores for each of the original answers - and then choose the one with the highest probability as our predicted answer.

4.1. Dataset alterations

In the original paper, each input sequence consisted of a classifier token ('[CLS]'), the question, a separator token ('[SEP]'), a possible answer, and then another separator token. The classifier token is unique, as the final hidden state corresponding to that token was used as the aggregate sequence representation for the question-answering task. Our model deviates from these assumptions. Instead, each input sequence was constructed from the question and 2 possible answers in the following manner: A classifier token, the first answer, a separator token, the question, a separator token, the second answer, and another classifier token. Note, that this construction break several of the original BERT paper assumptions - each of our input sequences consist of more than just one classifier token, and more than just two sentences. As such, it is possible that BERT will suffer from reduced performance, as its entire training process was built around these assumptions. However, breaking these assumptions was a crucial part of what made our constructions work - it allowed for each of the classifier tokens to "absorb" the meaning of the answer closer to it, and imporved the symmetry of our model in regard to each of the two answers.

Another part of the input sequence given to BERT is the segment embeddings - a number from 0, 1 which signifies wether the current token is part of the first sentece or the second. After trial and error, it was found that we perform best when giving a value of 1 for **each** of the answer tokens, and 0 for the question tokens. That makes sense, as part of the desire for symmetry in our model, and the fact that BERT wasn't trained with a possibilty for a third sentence, so using, for example, the number 2 for our second answer would not make any sense to it.

After passing our input sequence through BERT, we use a concatenation of the final hidden states matching the classifier tokens as the sequence representation. We learn as a parameter a vector whose dot product with the sequence representation is the score for the given answer pair.

4.2. Pair scores integration

We conceived a few methods of combining the different pair scores into probabilities for the answers. In this section we will review those methods and compare them.

4.2.1. Learnt weights

In this method, we introduce more layers to derive the indiviual probabilities from the pair scores. We used 2 layers, with a hidden size of $(number\ of\ pairs)^2$ and normalizing with a softmax layer.

4.2.2. Markov chains

In this method, we find the stationary distribution of a weighted graph whose edge's weights are derived from our pair scores. The reasoning for this process is as follows - imagine a graph whose vertices are the given answers. We begin by choosing an answer randomly, and we want to move on to the best possible answer. At each time step, we move to a different answer, choosing based on the scores of the pairs in which our current answer participates, i.e. the better another answer performed against our current answer, the more likely we are to move to it. This process is powerful, because it captures the dependency between the different answers, e.g. if answer B is better than answer A, and answer C is better than answer B, than we'd like to factor in that information when comparing between answers A and C.

Simply constructing a matrix with each cell being the score for the matching pair, normalizing each row so that we have a probability matrix, and then calculating the stationary distribution doesn't work well in practice. This is because normalizing the matrix causes the model to be unable to capture the absolute value of the scores, e.g. if it sees a row in which all values are similar, it can't differ the case in which all scores were low (and thus our "current" answer is good) from the case all of the scores were high. Consider the following matrix:

$$\begin{bmatrix} 0 & 0.95 & 0.85 & 0.25 \\ 0.05 & 0 & 0.98 & 0.74 \\ 0.15 & 0.02 & 0 & 0.45 \\ 0.75 & 0.26 & 0.55 & 0 \end{bmatrix}$$

We can see that the third answer had won in every comparison - thus, we would expect it to be assigned the highest probability. Given the currently described method, this in not the case, as the probabilities received would be:

$$\begin{bmatrix} 0.22614831 & 0.16630443 & 0.29576885 & \mathbf{0.31177841} \end{bmatrix}$$

We solve this problem by adding the average of each column to the matching cell on the diagonal. This makes sense, as this gives us the option of staying on the current choice, and the better a certain choice had performed against the other choices, the greater the possibilty of staying on it. After doing this, we derive this matrix for our example:

$$\begin{bmatrix} \mathbf{0.31} & 0.95 & 0.85 & 0.25 \\ 0.05 & \mathbf{0.41} & 0.98 & 0.74 \\ 0.15 & 0.02 & \mathbf{0.79} & 0.45 \\ 0.75 & 0.26 & 0.55 & \mathbf{0.48} \end{bmatrix}$$

And this probability matrix for our choices:

$$\begin{bmatrix} 0.16845236 & 0.13252954 & \mathbf{0.43521662} & 0.26380148 \end{bmatrix}$$

4.2.3. TIE-BREAKERS

This is not a method on its own, but it can be used on top of the other ones. Essentially, we can assert a "Certainty Threshold" on the probabilities obtained from the other methods, a rate between two probabilities under which we consider our prediction to be uncertain/unstable. If such a scenario arises, we then resort to choosing between the two highest-scored choices directly - that is, by looking at the score given to the pair and deciding accordingly.

Note that this change does not cause a change in the probabilities themselves, it only changes the predicted answer. As such, it does not cause a change in the measured loss, thus does not affect the learned parameters. It does change, however, the measured accuracy, which means that when learning parameters we will favor a parameter set that maximizes the accuracy given this change.

5. Experiments and Results

As our model deviates greatly from the assumptions that were made when the original BERT weights were trained, it is understanable that our model required more epochs to converge than the amount of epochs the original paper used when fine-tuning to the question answering task. We should also note that because each sample now consists of much more input sequences (the original model had one sequence per possible answer, our model has one sequence per possible answer **pair**), we are now more limited in our batch size, as the GPU has to hold the entire batch simultaneously. This problem can be mitigated by using gradient accumulation.

Our models were trained in parts. In the first part, we trained solely on the dataset alterations part, without intergrating the pair scores. That is, our model was trained directly on identifying the better answer in a pair

6. Stuff from the example paper

Submission citeTechReport to ICML 2019 will be entirely electronic, via a web site (not email). Information about the submission process and LaTeX templates are available on the conference web site at:

`http://icml.cc/`

The guidelines below will be enforced for initial submissions and camera-ready copies. Here is a brief summary:

- Submissions must be in PDF.
- Submitted papers can be up to eight pages long, not including references, and up to twelve pages when references and acknowledgments are included. Any paper exceeding this length will automatically be rejected.
- **Do not include author information or acknowledgements** in your initial submission.
- Your paper should be in **10 point Times font**.
- Make sure your PDF file only uses Type-1 fonts.
- Place figure captions *under* the figure (and omit titles from inside the graphic file itself). Place table captions *over* the table.
- References must include page numbers whenever possible and be as complete as possible. Place multiple citations in chronological order.
- Do not alter the style template; in particular, do not compress the paper format by reducing the vertical spaces.
- Keep your abstract brief and self-contained, one paragraph and roughly 4–6 sentences. Gross violations will require correction at the camera-ready phase. The title should have content words capitalized.

6.1. Submitting Papers

Paper Deadline: The deadline for paper submission that is advertised on the conference website is strict. If your full, anonymized, submission does not reach us on time, it will not be considered for publication.

Anonymous Submission: ICML uses double-blind review: no identifying author information may appear on the title page or in the paper itself. Section 7.3 gives further details.

Simultaneous Submission: ICML will not accept any paper which, at the time of submission, is under review for another conference or has already been published. This policy also applies to papers that overlap substantially in technical content with conference papers under review or

previously published. ICML submissions must not be submitted to other conferences during ICML's review period. Authors may submit to ICML substantially different versions of journal papers that are currently under review by the journal, but not yet accepted at the time of submission. Informal publications, such as technical reports or papers in workshop proceedings which do not appear in print, do not fall under these restrictions.

Authors must provide their manuscripts in **PDF** format. Furthermore, please make sure that files contain only embedded Type-1 fonts (e.g., using the program `pdffonts` in linux or using File/DocumentProperties/Fonts in Acrobat). Other fonts (like Type-3) might come from graphics files imported into the document.

Authors using **Word** must convert their document to PDF. Most of the latest versions of Word have the facility to do this automatically. Submissions will not be accepted in Word format or any format other than PDF. Really. We're not joking. Don't send Word.

Those who use **LaTeX** should avoid including Type-3 fonts. Those using `latex` and `dvips` may need the following two commands:

```
dvips -Ppdf -tletter -G0 -o paper.ps paper.dvi
ps2pdf paper.ps
```

It is a zero following the "-G", which tells dvips to use the config.pdf file. Newer TeX distributions don't always need this option.

Using `pdflatex` rather than `latex`, often gives better results. This program avoids the Type-3 font problem, and supports more advanced features in the `microtype` package.

Graphics files should be a reasonable size, and included from an appropriate format. Use vector formats (.eps/.pdf) for plots, lossless bitmap formats (.png) for raster graphics with sharp lines, and jpeg for photo-like images.

The style file uses the `hyperref` package to make clickable links in documents. If this causes problems for you, add `nohyperref` as one of the options to the `icml2019` usepackage statement.

6.2. Submitting Final Camera-Ready Copy

The final versions of papers accepted for publication should follow the same format and naming convention as initial submissions, except that author information (names and affiliations) should be given. See Section 7.3.2 for formatting instructions.

The footnote, "Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute." must be modified to "*Proceedings of the 36^{th} International Conference on Machine Learning*, Long Beach, USA, 2019. Copyright 2019 by the author(s)."

For those using the **LaTeX** style file, this change (and others) is handled automatically by simply changing `\usepackage{icml2019}` to

```
\usepackage[accepted]{icml2019}
```

Authors using **Word** must edit the footnote on the first page of the document themselves.

Camera-ready copies should have the title of the paper as running head on each page except the first one. The running title consists of a single line centered above a horizontal rule which is 1 point thick. The running head should be centered, bold and in 9 point type. The rule should be 10 points above the main text. For those using the **LaTeX** style file, the original title is automatically set as running head using the `fancyhdr` package which is included in the ICML 2019 style file package. In case that the original title exceeds the size restrictions, a shorter form can be supplied by using

```
\icmltitlerunning{...}
```

just before `\begin{document}`. Authors using **Word** must edit the header of the document themselves.

7. Format of the Paper

All submissions must follow the specified format.

7.1. Length and Dimensions

Submitted papers can be up to eight pages long, not including references, and up to twelve pages when references and acknowledgments are included. Acknowledgements should be limited to grants and people who contributed to the paper. Any submission that exceeds this page limit, or that diverges significantly from the specified format, will be rejected without review.

The text of the paper should be formatted in two columns, with an overall width of 6.75 inches, height of 9.0 inches, and 0.25 inches between the columns. The left margin should be 0.75 inches and the top margin 1.0 inch (2.54 cm). The right and bottom margins will depend on whether you print on US letter or A4 paper, but all final versions must be produced for US letter size.

The paper body should be set in 10 point type with a vertical spacing of 11 points. Please use Times typeface throughout the text.

7.2. Title

The paper title should be set in 14 point bold type and centered between two horizontal rules that are 1 point thick,

with 1.0 inch between the top rule and the top edge of the page. Capitalize the first letter of content words and put the rest of the title in lower case.

7.3. Author Information for Submission

ICML uses double-blind review, so author information must not appear. If you are using LaTeX and the `icml2019.sty` file, use `\icmlauthor{...}` to specify authors and `\icmlaffiliation{...}` to specify affiliations. (Read the TeX code used to produce this document for an example usage.) The author information will not be printed unless `accepted` is passed as an argument to the style file. Submissions that include the author information will not be reviewed.

7.3.1. Self-Citations

If you are citing published papers for which you are an author, refer to yourself in the third person. In particular, do not use phrases that reveal your identity (e.g., "in previous work (**?**), we have shown ...").

Do not anonymize citations in the reference section. The only exception are manuscripts that are not yet published (e.g., under submission). If you choose to refer to such unpublished manuscripts (**?**), anonymized copies have to be submitted as Supplementary Material via CMT. However, keep in mind that an ICML paper should be self contained and should contain sufficient detail for the reviewers to evaluate the work. In particular, reviewers are not required to look at the Supplementary Material when writing their review.

7.3.2. Camera-Ready Author Information

If a paper is accepted, a final camera-ready copy must be prepared. For camera-ready papers, author information should start 0.3 inches below the bottom rule surrounding the title. The authors' names should appear in 10 point bold type, in a row, separated by white space, and centered. Author names should not be broken across lines. Unbolded superscripted numbers, starting 1, should be used to refer to affiliations.

Affiliations should be numbered in the order of appearance. A single footnote block of text should be used to list all the affiliations. (Academic affiliations should list Department, University, City, State/Region, Country. Similarly for industrial affiliations.)

Each distinct affiliations should be listed once. If an author has multiple affiliations, multiple superscripts should be placed after the name, separated by thin spaces. If the authors would like to highlight equal contribution by multiple first authors, those authors should have an asterisk placed after their name in superscript, and the term "*Equal contribution" should be placed in the footnote block ahead of the list of affiliations. A list of corresponding authors and their emails (in the format Full Name <email@domain.com>) can follow the list of affiliations. Ideally only one or two names should be listed.

A sample file with author names is included in the ICML2019 style file package. Turn on the `[accepted]` option to the stylefile to see the names rendered. All of the guidelines above are implemented by the LaTeX style file.

7.4. Abstract

The paper abstract should begin in the left column, 0.4 inches below the final address. The heading 'Abstract' should be centered, bold, and in 11 point type. The abstract body should use 10 point type, with a vertical spacing of 11 points, and should be indented 0.25 inches more than normal on left-hand and right-hand margins. Insert 0.4 inches of blank space after the body. Keep your abstract brief and self-contained, limiting it to one paragraph and roughly 4–6 sentences. Gross violations will require correction at the camera-ready phase.

7.5. Partitioning the Text

You should organize your paper into sections and paragraphs to help readers place a structure on the material and understand its contributions.

7.5.1. Sections and Subsections

Section headings should be numbered, flush left, and set in 11 pt bold type with the content words capitalized. Leave 0.25 inches of space before the heading and 0.15 inches after the heading.

Similarly, subsection headings should be numbered, flush left, and set in 10 pt bold type with the content words capitalized. Leave 0.2 inches of space before the heading and 0.13 inches afterward.

Finally, subsubsection headings should be numbered, flush left, and set in 10 pt small caps with the content words capitalized. Leave 0.18 inches of space before the heading and 0.1 inches after the heading.

Please use no more than three levels of headings.

7.5.2. Paragraphs and Footnotes

Within each section or subsection, you should further partition the paper into paragraphs. Do not indent the first line of a given paragraph, but insert a blank line between succeeding ones.

You can use footnotes[1] to provide readers with additional information about a topic without interrupting the flow of the paper. Indicate footnotes with a number in the text where the point is most relevant. Place the footnote in 9 point type at the bottom of the column in which it appears. Precede the first footnote in a column with a horizontal rule of 0.8 inches.[2]

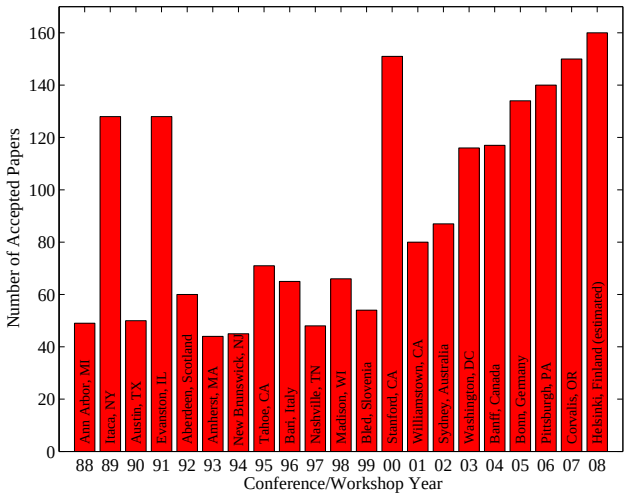

Figure 2. Historical locations and number of accepted papers for International Machine Learning Conferences (ICML 1993 – ICML 2008) and International Workshops on Machine Learning (ML 1988 – ML 1992). At the time this figure was produced, the number of accepted papers for ICML 2008 was unknown and instead estimated.

7.6. Figures

You may want to include figures in the paper to illustrate your approach and results. Such artwork should be centered, legible, and separated from the text. Lines should be dark and at least 0.5 points thick for purposes of reproduction, and text should not appear on a gray background.

Label all distinct components of each figure. If the figure takes the form of a graph, then give a name for each axis and include a legend that briefly describes each curve. Do not include a title inside the figure; instead, the caption should serve this function.

Number figures sequentially, placing the figure number and caption *after* the graphics, with at least 0.1 inches of space before the caption and 0.1 inches after it, as in Figure 2. The figure caption should be set in 9 point type and centered unless it runs two or more lines, in which case it should be flush left. You may float figures to the top or bottom of a column, and you may set wide figures across both columns (use the environment `figure*` in LaTeX). Always place two-column figures at the top or bottom of the page.

[1]Footnotes should be complete sentences.

[2]Multiple footnotes can appear in each column, in the same order as they appear in the text, but spread them across columns and pages if possible.

Algorithm 1 Bubble Sort

Input: data x_i, size m
repeat
  Initialize $noChange = true$.
  for $i = 1$ **to** $m - 1$ **do**
    if $x_i > x_{i+1}$ **then**
      Swap x_i and x_{i+1}
      $noChange = false$
    end if
  end for
until $noChange$ is $true$

Table 1. Classification accuracies for naive Bayes and flexible Bayes on various data sets.

DATA SET	NAIVE	FLEXIBLE	BETTER?
BREAST	95.9± 0.2	96.7± 0.2	√
CLEVELAND	83.3± 0.6	80.0± 0.6	×
GLASS2	61.9± 1.4	83.8± 0.7	√
CREDIT	74.8± 0.5	78.3± 0.6	
HORSE	73.3± 0.9	69.7± 1.0	×
META	67.1± 0.6	76.5± 0.5	√
PIMA	75.1± 0.6	73.9± 0.5	
VEHICLE	44.9± 0.6	61.5± 0.4	√

7.7. Algorithms

If you are using LaTeX, please use the "algorithm" and "algorithmic" environments to format pseudocode. These require the corresponding stylefiles, algorithm.sty and algorithmic.sty, which are supplied with this package. Algorithm 1 shows an example.

7.8. Tables

You may also want to include tables that summarize material. Like figures, these should be centered, legible, and numbered consecutively. However, place the title *above* the table with at least 0.1 inches of space before the title and the same after it, as in Table 1. The table title should be set in 9 point type and centered unless it runs two or more lines, in which case it should be flush left.

Tables contain textual material, whereas figures contain graphical material. Specify the contents of each row and column in the table's topmost row. Again, you may float tables to a column's top or bottom, and set wide tables across both columns. Place two-column tables at the top or bottom of the page.

7.9. Citations and References

Please use APA reference format regardless of your formatter or word processor. If you rely on the LaTeX bibliographic facility, use `natbib.sty` and `icml2019.bst` included in the style-file package to obtain this format.

Citations within the text should include the authors' last names and year. If the authors' names are included in the sentence, place only the year in parentheses, for example when referencing Arthur Samuel's pioneering work (**?**). Otherwise place the entire reference in parentheses with the authors and year separated by a comma (**?**). List multiple references separated by semicolons (**???**). Use the 'et al.' construct only for citations with three or more authors or after listing all authors to a publication in an earlier reference (**?**).

Authors should cite their own work in the third person in the initial version of their paper submitted for blind review. Please refer to Section 7.3 for detailed instructions on how to cite your own papers.

Use an unnumbered first-level section heading for the references, and use a hanging indent style, with the first line of the reference flush against the left margin and subsequent lines indented by 10 points. The references at the end of this document give examples for journal articles (**?**), conference publications (**?**), book chapters (**?**), books (**?**), edited volumes (**?**), technical reports (**?**), and dissertations (**?**).

Alphabetize references by the surnames of the first authors, with single author entries preceding multiple author entries. Order references for the same authors by year of publication, with the earliest first. Make sure that each reference includes all relevant information (e.g., page numbers).

Please put some effort into making references complete, presentable, and consistent. If using bibtex, please protect capital letters of names and abbreviations in titles, for example, use {B}ayesian or {L}ipschitz in your .bib file.

7.10. Software and Data

We strongly encourage the publication of software and data with the camera-ready version of the paper whenever appropriate. This can be done by including a URL in the camera-ready copy. However, do not include URLs that reveal your institution or identity in your submission for review. Instead, provide an anonymous URL or upload the material as "Supplementary Material" into the CMT reviewing system. Note that reviewers are not required to look at this material when writing their review.

Acknowledgements

Do not include acknowledgements in the initial version of the paper submitted for blind review.

If a paper is accepted, the final camera-ready version can (and probably should) include acknowledgements. In this case, please place such acknowledgements in an unnumbered section at the end of the paper. Typically, this will include thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

Storks, S., Gao, Q., and Chai, J. Y. Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*, 2019.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*, 2018.

A. Do *not* have an appendix here

Do not put content after the references. Put anything that you might normally include after the references in a separate supplementary file.

We recommend that you build supplementary material in a separate document. If you must create one PDF and cut it up, please be careful to use a tool that doesn't alter the margins, and that doesn't aggressively rewrite the PDF file. pdftk usually works fine.

Please do not use Apple's preview to cut off supplementary material. In previous years it has altered margins, and created headaches at the camera-ready stage.