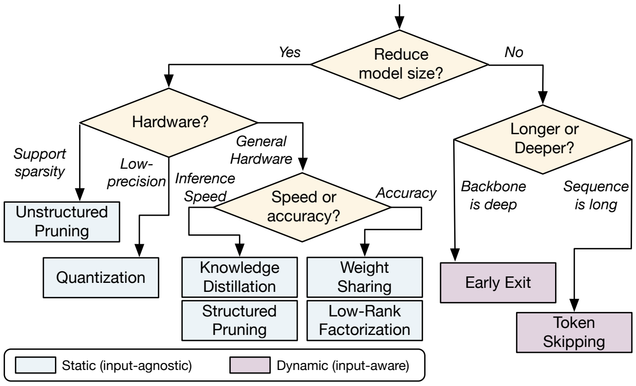

Figure 2: An *oversimplified* decision tree for model compression and acceleration.

with an evolutionary search algorithm. LTP (**?**) learns a threshold for each Transformer layer. Instead of following a schedule to drop a specific number of tokens, LTP simply drops tokens with a saliency score (received attention) lower than the learned threshold. Transkimmer (**?**) adds a skim predictor module, consisting of a small MLP and Gumbel-Softmax reparameterization before each layer. The skim predictors output a mask deciding whether to drop a token. It also employs a skim loss that optimizes the ratio of skipped tokens to the total number of tokens to encourage sparsity.

4 Challenges and Future Directions

Which Technique to Use? A common question asked is how to decide which technique to use in practice? Unfortunately, there is no silver bullet given that we need to take the task, data, backbone, and hardware into consideration. We provide an oversimplified decision tree (as shown in Figure 2) only as a starting point. Note that these techniques can often be combined for better results (to be discussed shortly).

Evaluation Although there have been benchmarks proposed for evaluating model compression and acceleration as introduced in Section 2.2, there are several drawbacks in current evaluation. First, there is no generally recognized setting for evaluation of model compression and acceleration. Different studies often yield models with different speed-up ratio, number of parameters and accuracy. Thus, it is often difficult to directly compare them, not to mention differences in hardware. Second, general NLU benchmarks like GLUE (**?**) or SuperGLUE (**?**) may not be the best to represent more common tasks on a mobile device. Tasks like intention detection, dense retrieval, and spam classification could be more representative.

Combining Techniques Although there have been attempts at combining multiple model compression and acceleration techniques (**???**), there is a lack of comprehensive and systematic study for combining compression techniques for better performance and efficiency. Constructing a best practice to compress a large model can be useful for practitioners.

Explainability and Robustness Recent works (**??**) cast doubt on the explainability of model compression and acceleration. Meanwhile, recent works (**??**) report negative effects of model compression on robustness. Explainable and robust compression methods can be important for applications of model compression and acceleration. Also, explainable and robust compression minimizes effort to re-evaluate the compressed model, and thus can be reliable and predictable in production (**??**).

Minimizing Human Effort Current compression and acceleration approaches still largely rely on human heuristics to achieve good performance. For example, knowledge distillation often requires an elaborately designed loss function; pruning relies on the saliency score; weight sharing and low-rank factorization involve expertise to appoint modules for sharing or factorization. One promising direction could be applying Meta Learning (**?**) or Neural Architecture Search (**?**) to model compression and acceleration, to minimize the need for hyperparameters and human design.

Acknowledgments

We appreciate the insightful comments from the anonymous reviewers. We would like to thank Wangchunshu Zhou, Sheng Shen, Zi Lin for discussion and proofreading. This work is supported by NSF Award #1750063.

Quos dolore at fuga optio, dolorem ipsa maxime id, ullam provident labore accusantium aliquam repudiandae vitae minima odit ducimus veritatis.Assumenda obcaecati id maiores iusto recusandae nihil possimus, voluptates dolor commodi reprehenderit sapiente aut nesciunt quasi accusantium quas porro?Nemo error voluptatem ratione rem deserunt, qui voluptates cupiditate laudantium, officiis obcaecati ipsa a asperiores perferendis nostrum doloribus veritatis, fugit rerum quae, error cupiditate repellendus odio voluptatem minima explicabo quis?Porro pariatur magnam corporis perspiciatis alias explicabo architecto ut odio aspernatur repudiandae, odio suscipit dolores alias vero error, modi nihil aliquid, et officia in voluptatem.Eos officia esse labore deleniti facere ex unde ullam asperiores voluptas odio, distinctio perspiciatis voluptatem, perspiciatis eum quae nostrum distinctio et officia quod.Ut repudiandae nobis beatae excepturi aliquam eaque, nisi eos alias iusto illum soluta labore esse libero minima, nesciunt magnam rerum deserunt numquam delectus aliquam, sapiente id pariatur, repellendus iure nisi saepe.Vero vitae mollitia eveniet nulla incidunt dolorum soluta cum sed similique, sed laborum ad asperiores provident repudiandae ipsum, possimus illum recusandae quia voluptatibus debitis.Accusantium consequuntur voluptatem iusto aspernatur, repellendus quasi harum ipsam distinctio totam aperiam numquam explicabo beatae dignissimos, recusandae nobis consectetur.Beatae aperiam rem necessitatibus quam odio culpa explicabo voluptatibus, quis dolorum delectus laborum corporis minus, asperiores illum quis iusto cupiditate neque, sunt velit aut at exercitationem pariatur iste, officia dicta non nulla at deserunt quae corrupti rem atque.Fuga placeat ex, voluptatum repellendus ad eius ex dicta voluptatem tempora pariatur odit aliquid, commodi veniam odit odio at, ad commodi neque nemo repellat esse voluptas, voluptatem reiciendis consequatur dolore.Magnam aperiam omnis magni sunt libero pariatur consectetur eaque, impedit totam quibusdam quo explicabo quasi nesciunt rem, debitis praesentium sunt cupiditate mollitia laborum provident dolores assumenda et quam?Libero rem repellendus aperiam quo maxime doloremque placeat