Adaptively Sparse Transformers

Gonçalo M. Correia[ℵ] goncalo.correia@lx.it.pt
Vlad Niculae[ℵ] vlad@vene.ro
André F.T. Martins[ℵ][ב] andre.martins@unbabel.com

[ℵ]Instituto de Telecomunicações, Lisbon, Portugal
[ב]Unbabel, Lisbon, Portugal

Abstract

Attention mechanisms have become ubiquitous in NLP. Recent architectures, notably the Transformer, learn powerful context-aware word representations through layered, multi-headed attention. The multiple heads learn diverse types of word relationships. However, with standard softmax attention, all attention heads are dense, assigning a non-zero weight to all context words. In this work, we introduce the adaptively sparse Transformer, wherein attention heads have flexible, context-dependent sparsity patterns. This sparsity is accomplished by replacing softmax with α-entmax: a differentiable generalization of softmax that allows low-scoring words to receive precisely zero weight. Moreover, we derive a method to automatically learn the α parameter – which controls the shape and sparsity of α-entmax – allowing attention heads to choose between focused or spread-out behavior. Our adaptively sparse Transformer improves interpretability and head diversity when compared to softmax Transformers on machine translation datasets. Findings of the quantitative and qualitative analysis of our approach include that heads in different layers learn different sparsity preferences and tend to be more diverse in their attention distributions than softmax Transformers. Furthermore, at no cost in accuracy, sparsity in attention heads helps to uncover different head specializations.

1 Introduction

The Transformer architecture (Vaswani et al., 2017) for deep neural networks has quickly risen to prominence in NLP through its efficiency and performance, leading to improvements in the state of the art of Neural Machine Translation (NMT; Junczys-Dowmunt et al., 2018; Ott et al., 2018), as well as inspiring other powerful general-purpose models like BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019). At the heart of the Transformer lie *multi-head attention* mechanisms: each word is represented by multiple different weighted averages of its relevant context. As suggested by recent works on interpreting attention head roles, separate attention heads may learn to look for various relationships between tokens (Tang et al., 2018; Raganato and Tiedemann, 2018; Mareček and Rosa, 2018; Tenney et al., 2019; Voita et al., 2019).

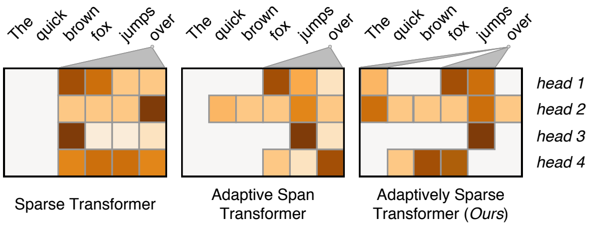

Figure 1: Attention distributions of different self-attention heads for the time step of the token “over”, shown to compare our model to other related work. While the sparse Transformer (Child et al., 2019) and the adaptive span Transformer (Sukhbaatar et al., 2019) only attend to words within a contiguous span of the past tokens, our model is not only able to obtain different and not necessarily contiguous sparsity patterns for each attention head, but is also able to tune its support over which tokens to attend adaptively.

The attention distribution of each head is predicted typically using the **softmax** normalizing transform. As a result, all context words have non-zero attention weight. Recent work on single attention architectures suggest that using sparse normalizing transforms in attention mechanisms such as sparsemax – which can yield exactly zero probabilities for irrelevant words – may improve performance and interpretability (Malaviya et al., 2018; Deng et al., 2018; Peters et al., 2019). Qualitative analysis of attention heads (Vaswani et al., 2017, Figure 5) suggests that, depending on what phenomena they capture, heads tend to favor flatter or more peaked distributions.

Recent works have proposed sparse Transform-

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 2174–2184,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

ers (Child et al., 2019) and adaptive span Transformers (Sukhbaatar et al., 2019). However, the "sparsity" of those models only limits the attention to a contiguous span of past tokens, while in this work we propose a **highly adaptive** Transformer model that is capable of attending to a sparse set of words that are not necessarily contiguous. Figure 1 shows the relationship of these methods with ours.

Our contributions are the following:

- We introduce **sparse attention** into the Transformer architecture, showing that it eases interpretability and leads to slight accuracy gains.
- We propose an adaptive version of sparse attention, where the shape of each attention head is **learnable** and can vary continuously and dynamically between the dense limit case of *softmax* and the sparse, piecewise-linear *sparsemax* case.[1]
- We make an extensive analysis of the added interpretability of these models, identifying both crisper examples of attention head behavior observed in previous work, as well as novel behaviors unraveled thanks to the sparsity and adaptivity of our proposed model.

2 Background

2.1 The Transformer

In NMT, the Transformer (Vaswani et al., 2017) is a sequence-to-sequence (seq2seq) model which maps an input sequence to an output sequence through hierarchical **multi-head attention** mechanisms, yielding a dynamic, context-dependent strategy for propagating information within and across sentences. It contrasts with previous seq2seq models, which usually rely either on costly gated recurrent operations (often LSTMs: Bahdanau et al., 2015; Luong et al., 2015) or static convolutions (Gehring et al., 2017).

Given n query contexts and m sequence items under consideration, attention mechanisms compute, for each query, a weighted representation of the items. The particular attention mechanism used in Vaswani et al. (2017) is called *scaled dot-product attention*, and it is computed in the following way:

$$\mathsf{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{\pi}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}}\right)\boldsymbol{V}, \qquad (1)$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times d}$ contains representations of the queries, $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{m\times d}$ are the *keys* and *values* of the items attended over, and d is the dimensionality of these representations. The $\boldsymbol{\pi}$ mapping normalizes row-wise using **softmax**, $\boldsymbol{\pi}(\boldsymbol{Z})_{ij} = \mathsf{softmax}(\boldsymbol{z}_i)_j$, where

$$\mathsf{softmax}(\boldsymbol{z}) = \frac{\exp(z_j)}{\sum_{j'} \exp(z_{j'})}. \qquad (2)$$

In words, the *keys* are used to compute a relevance score between each item and query. Then, normalized attention weights are computed using softmax, and these are used to weight the *values* of each item at each query context.

However, for complex tasks, different parts of a sequence may be relevant in different ways, motivating *multi-head attention* in Transformers. This is simply the application of Equation 1 in parallel H times, each with a different, learned linear transformation that allows specialization:

$$\mathsf{Head}_i(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathsf{Att}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V) \qquad (3)$$

In the Transformer, there are three separate multi-head attention mechanisms for distinct purposes:

- **Encoder self-attention:** builds rich, layered representations of each input word, by attending on the entire input sentence.
- **Context attention:** selects a representative weighted average of the encodings of the input words, at each time step of the decoder.
- **Decoder self-attention:** attends over the partial output sentence fragment produced so far.

Together, these mechanisms enable the contextualized flow of information between the input sentence and the sequential decoder.

2.2 Sparse Attention

The softmax mapping (Equation 2) is elementwise proportional to exp, therefore it can never assign a weight of **exactly zero**. Thus, unnecessary items are still taken into consideration to some extent. Since its output sums to one, this invariably means less weight is assigned to the relevant items, potentially harming performance and interpretability (Jain and Wallace, 2019). This has motivated a line of research on learning networks with *sparse* mappings (Martins and Astudillo, 2016; Niculae and Blondel, 2017; Louizos et al., 2018; Shao et al.,

[1] Code and pip package available at `https://github.com/deep-spin/entmax`.

2019). We focus on a recently-introduced flexible family of transformations, α-entmax (Blondel et al., 2019; Peters et al., 2019), defined as:

$$\alpha\text{-entmax}(\boldsymbol{z}) := \underset{\boldsymbol{p}\in\triangle^d}{\operatorname{argmax}}\, \boldsymbol{p}^\top \boldsymbol{z} + \mathsf{H}^\mathsf{T}_\alpha(\boldsymbol{p}), \qquad (4)$$

where $\triangle^d := \{\boldsymbol{p} \in \mathbb{R}^d : \sum_i p_i = 1\}$ is the *probability simplex*, and, for $\alpha \geq 1$, $\mathsf{H}^\mathsf{T}_\alpha$ is the Tsallis continuous family of entropies (Tsallis, 1988):

$$\mathsf{H}^\mathsf{T}_\alpha(\boldsymbol{p}) := \begin{cases} \frac{1}{\alpha(\alpha-1)} \sum_j \left(p_j - p_j^\alpha\right), & \alpha \neq 1, \\ -\sum_j p_j \log p_j, & \alpha = 1. \end{cases} \qquad (5)$$

This family contains the well-known Shannon and Gini entropies, corresponding to the cases $\alpha = 1$ and $\alpha = 2$, respectively.

Equation 4 involves a convex optimization subproblem. Using the definition of $\mathsf{H}^\mathsf{T}_\alpha$, the optimality conditions may be used to derive the following form for the solution (Appendix B.2):

$$\alpha\text{-entmax}(\boldsymbol{z}) = [(\alpha - 1)\boldsymbol{z} - \tau\mathbf{1}]_+^{1/\alpha-1}, \qquad (6)$$

where $[\cdot]_+$ is the positive part (ReLU) function, $\mathbf{1}$ denotes the vector of all ones, and τ – which acts like a threshold – is the Lagrange multiplier corresponding to the $\sum_i p_i = 1$ constraint.

Properties of $\boldsymbol{\alpha}$-entmax. The appeal of α-entmax for attention rests on the following properties. For $\alpha = 1$ (*i.e.*, when $\mathsf{H}^\mathsf{T}_\alpha$ becomes the Shannon entropy), it exactly recovers the softmax mapping (We provide a short derivation in Appendix B.3.). For all $\alpha > 1$ it permits sparse solutions, in stark contrast to softmax. In particular, for $\alpha = 2$, it recovers the sparsemax mapping (Martins and Astudillo, 2016), which is piecewise linear. In-between, as α increases, the mapping continuously gets sparser as its curvature changes.

To compute the value of α-entmax, one must find the threshold τ such that the *r.h.s.* in Equation 6 sums to one. Blondel et al. (2019) propose a general bisection algorithm. Peters et al. (2019) introduce a faster, exact algorithm for $\alpha = 1.5$, and enable using α-entmax with fixed α within a neural network by showing that the α-entmax Jacobian *w.r.t.* $\boldsymbol{z}$ for $\boldsymbol{p}^\star = \alpha\text{-entmax}(\boldsymbol{z})$ is

$$\frac{\partial\, \alpha\text{-entmax}(\boldsymbol{z})}{\partial \boldsymbol{z}} = \operatorname{diag}(\boldsymbol{s}) - \frac{1}{\sum_j s_j}\boldsymbol{s}\boldsymbol{s}^\top, \quad \text{where} \quad s_i = \begin{cases} (p_i^\star)^{2-\alpha}, & p_i^\star > 0, \\ 0, & p_i^\star = 0. \end{cases} \qquad (7)$$

Our work furthers the study of α-entmax by providing a derivation of the Jacobian ***w.r.t.* the hyper-parameter $\boldsymbol{\alpha}$** (Section 3), thereby allowing the shape and sparsity of the mapping to be learned automatically. This is particularly appealing in the context of multi-head attention mechanisms, where we shall show in Section 5.1 that different heads tend to learn different sparsity behaviors.

3 Adaptively Sparse Transformers with $\boldsymbol{\alpha}$-entmax

We now propose a novel Transformer architecture wherein we simply replace softmax with α-entmax in the attention heads. Concretely, we replace the row normalization $\boldsymbol{\pi}$ in Equation 1 by

$$\boldsymbol{\pi}(\boldsymbol{Z})_{ij} = \alpha\text{-entmax}(\boldsymbol{z}_i)_j \qquad (8)$$

This change leads to sparse attention weights, as long as $\alpha > 1$; in particular, $\alpha = 1.5$ is a sensible starting point (Peters et al., 2019).

Different $\boldsymbol{\alpha}$ per head. Unlike LSTM-based seq2seq models, where α can be more easily tuned by grid search, in a Transformer, there are many attention heads in multiple layers. Crucial to the power of such models, the different heads capture different linguistic phenomena, some of them isolating important words, others spreading out attention across phrases (Vaswani et al., 2017, Figure 5). This motivates using different, adaptive α values for each attention head, such that some heads may learn to be sparser, and others may become closer to softmax. We propose doing so by treating the α values as neural network parameters, optimized via stochastic gradients along with the other weights.

Derivatives *w.r.t.* $\boldsymbol{\alpha}$. In order to optimize α automatically via gradient methods, we must compute the Jacobian of the entmax output *w.r.t.* α. Since entmax is defined through an optimization problem, this is non-trivial and cannot be simply handled through automatic differentiation; it falls within the domain of *argmin differentiation*, an active research topic in optimization (Gould et al., 2016; Amos and Kolter, 2017).

One of our key contributions is the derivation of a closed-form expression for this Jacobian. The next proposition provides such an expression, enabling entmax layers with adaptive α. To the best of our knowledge, ours is the first neural network module that can automatically, continuously vary in shape away from softmax and toward sparse mappings like sparsemax.

Proposition 1. *Let* $\boldsymbol{p}^\star := \alpha\text{-}\mathsf{entmax}(\boldsymbol{z})$ *be the solution of Equation 4. Denote the distribution* $\tilde{p}_i := {}^{(p_i^\star)^{2-\alpha}}/_{\sum_j (p_j^\star)^{2-\alpha}}$ *and let* $h_i := -p_i^\star \log p_i^\star$. *The* i^{th} *component of the Jacobian* $\boldsymbol{g} := \frac{\partial\, \alpha\text{-}\mathsf{entmax}(\boldsymbol{z})}{\partial \alpha}$ *is*

$$g_i = \begin{cases} \frac{p_i^\star - \tilde{p}_i}{(\alpha-1)^2} + \frac{h_i - \tilde{p}_i \sum_j h_j}{\alpha - 1}, & \alpha > 1, \\ \frac{h_i \log p_i^\star - p_i^\star \sum_j h_j \log p_j^\star}{2}, & \alpha = 1. \end{cases} \tag{9}$$

The proof uses implicit function differentiation and is given in Appendix C.

Proposition 1 provides the remaining missing piece needed for training adaptively sparse Transformers. In the following section, we evaluate this strategy on neural machine translation, and analyze the behavior of the learned attention heads.

4 Experiments

We apply our adaptively sparse Transformers on four machine translation tasks. For comparison, a natural baseline is the standard Transformer architecture using the softmax transform in its multi-head attention mechanisms. We consider two other model variants in our experiments that make use of different normalizing transformations:

- **1.5-entmax:** a Transformer with sparse entmax attention with fixed $\alpha = 1.5$ for all heads. This is a novel model, since 1.5-entmax had only been proposed for RNN-based NMT models (Peters et al., 2019), but never in Transformers, where attention modules are not just one single component of the seq2seq model but rather an integral part of all of the model components.

- **α-entmax:** an **adaptive** Transformer with sparse entmax attention with a different, learned $\alpha_{i,j}^t$ for each head.

The adaptive model has an additional scalar parameter per attention head per layer for each of the three attention mechanisms (encoder self-attention, context attention, and decoder self-attention), *i.e.*,

$$\begin{aligned} \{a_{i,j}^t \in \mathbb{R} :\ & i \in \{1, \ldots, L\}, j \in \{1, \ldots, H\}, \\ & t \in \{\mathtt{enc}, \mathtt{ctx}, \mathtt{dec}\}\}, \end{aligned} \tag{10}$$

and we set $\alpha_{i,j}^t = 1 + \mathsf{sigmoid}(a_{i,j}^t) \in]1, 2[$. All or some of the α values can be tied if desired, but we keep them independent for analysis purposes.

Datasets. Our models were trained on 4 machine translation datasets of different training sizes:

- IWSLT 2017 German → English (DE→EN, Cettolo et al., 2017): 200K sentence pairs.
- KFTT Japanese → English (JA→EN, Neubig, 2011): 300K sentence pairs.
- WMT 2016 Romanian → English (RO→EN, Bojar et al., 2016): 600K sentence pairs.
- WMT 2014 English → German (EN→DE, Bojar et al., 2014): 4.5M sentence pairs.

All of these datasets were preprocessed with byte-pair encoding (BPE; Sennrich et al., 2016), using joint segmentations of 32k merge operations.

Training. We follow the dimensions of the Transformer-Base model of Vaswani et al. (2017): The number of layers is $L = 6$ and number of heads is $H = 8$ in the encoder self-attention, the context attention, and the decoder self-attention. We use a mini-batch size of 8192 tokens and warm up the learning rate linearly until 20k steps, after which it decays according to an inverse square root schedule. All models were trained until convergence of validation accuracy, and evaluation was done at each 10k steps for RO→EN and EN→DE and at each 5k steps for DE→EN and JA→EN. The end-to-end computational overhead of our methods, when compared to standard softmax, is relatively small; in training tokens per second, the models using α-entmax and 1.5-entmax are, respectively, 75% and 90% the speed of the softmax model.

Results. We report test set tokenized BLEU (Papineni et al., 2002) results in Table 1. We can see that replacing softmax by entmax does not hurt performance in any of the datasets; indeed, sparse attention Transformers tend to have slightly higher BLEU, but their sparsity leads to a better potential for analysis. In the next section, we make use of this potential by exploring the learned internal mechanics of the self-attention heads.

5 Analysis

We conduct an analysis for the higher-resource dataset WMT 2014 English → German of the attention in the sparse adaptive Transformer model (α-entmax) at multiple levels: we analyze high-level statistics as well as individual head behavior. Moreover, we make a qualitative analysis of the interpretability capabilities of our models.

activation	DE→EN	JA→EN	RO→EN	EN→DE
softmax	29.79	21.57	32.70	26.02
1.5-entmax	29.83	**22.13**	**33.10**	25.89
α-entmax	**29.90**	21.74	32.89	**26.93**

Table 1: Machine translation tokenized BLEU test results on IWSLT 2017 DE→EN, KFTT JA→EN, WMT 2016 RO→EN and WMT 2014 EN→DE, respectively.

5.1 High-Level Statistics

What kind of α values are learned? Figure 2 shows the learning trajectories of the α parameters of a selected subset of heads. We generally observe a tendency for the randomly-initialized α parameters to decrease initially, suggesting that softmax-like behavior may be preferable while the model is still very uncertain. After around one thousand steps, some heads change direction and become sparser, perhaps as they become more confident and specialized. This shows that the initialization of α does not predetermine its sparsity level or the role the head will have throughout. In particular, head 8 in the encoder self-attention layer 2 first drops to around $\alpha = 1.3$ before becoming one of the sparsest heads, with $\alpha \approx 2$.

The overall distribution of α values at convergence can be seen in Figure 3. We can observe that the encoder self-attention blocks learn to concentrate the α values in two modes: a very sparse one around $\alpha \to 2$, and a dense one between softmax and 1.5-entmax. However, the decoder self and context attention only learn to distribute these parameters in a single mode. We show next that this is reflected in the average density of attention weight vectors as well.

Attention weight density when translating. For any $\alpha > 1$, it would still be possible for the weight matrices in Equation 3 to learn re-scalings so as to make attention sparser or denser. To visualize the impact of adaptive α values, we compare the empirical attention weight density (the average number of tokens receiving non-zero attention) within each module, against sparse Transformers with fixed $\alpha = 1.5$.

Figure 4 shows that, with fixed $\alpha = 1.5$, heads tend to be sparse and similarly-distributed in all three attention modules. With learned α, there are two notable changes: (i) a prominent mode corresponding to fully dense probabilities, showing that our models learn to combine sparse and dense attention, and (ii) a distinction between the encoder self-attention – whose background distribution tends toward extreme sparsity – and the other two modules, who exhibit more uniform background distributions. This suggests that perhaps entirely sparse Transformers are suboptimal.

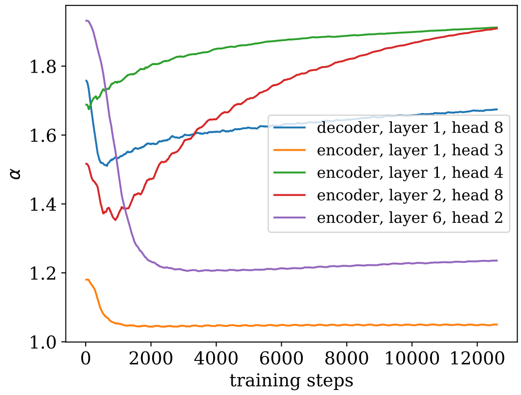

Figure 2: Trajectories of α values for a subset of the heads during training. Initialized at random, most heads become denser in the beginning, before converging. This suggests that dense attention may be more beneficial while the network is still uncertain, being replaced by sparse attention afterwards.

The fact that the decoder seems to prefer denser attention distributions might be attributed to it being auto-regressive, only having access to past tokens and not the full sentence. We speculate that it might lose too much information if it assigned weights of zero to too many tokens in the self-attention, since there are fewer tokens to attend to in the first place.

Teasing this down into separate layers, Figure 5 shows the average (sorted) density of each head for each layer. We observe that α-entmax is able to learn different sparsity patterns at each layer, leading to more variance in individual head behavior, to clearly-identified dense and sparse heads, and overall to different tendencies compared to the fixed case of $\alpha = 1.5$.

Head diversity. To measure the overall disagreement between attention heads, as a measure of head

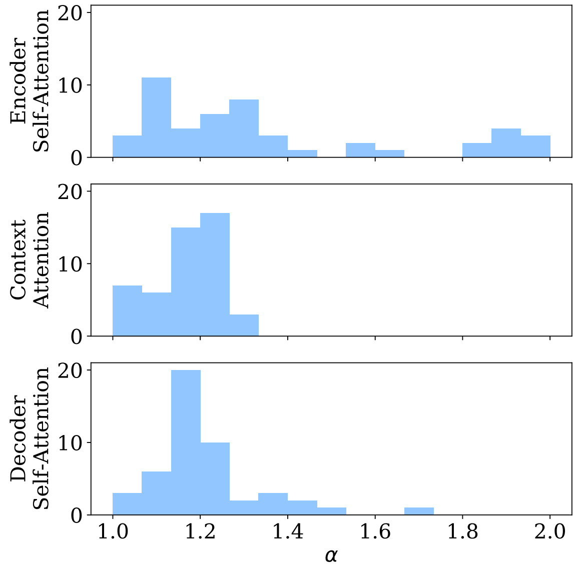

Figure 3: Distribution of learned α values per attention block. While the encoder self-attention has a bimodal distribution of values of α, the decoder self-attention and context attention have a single mode.

diversity, we use the following generalization of the Jensen-Shannon divergence:

$$JS = \mathsf{H}^{\mathrm{S}}\left(\frac{1}{H}\sum_{j=1}^{H} \boldsymbol{p}_j\right) - \frac{1}{H}\sum_{j=1}^{H} \mathsf{H}^{\mathrm{S}}(\boldsymbol{p}_j) \quad (11)$$

where $\boldsymbol{p}_j$ is the vector of attention weights assigned by head j to each word in the sequence, and H^{S} is the Shannon entropy, base-adjusted based on the dimension of $\boldsymbol{p}$ such that $JS \leq 1$. We average this measure over the entire validation set. The higher this metric is, the more the heads are taking different roles in the model.

Figure 6 shows that both sparse Transformer variants show more diversity than the traditional softmax one. Interestingly, diversity seems to peak in the middle layers of the encoder self-attention and context attention, while this is not the case for the decoder self-attention.

The statistics shown in this section can be found for the other language pairs in Appendix A.

5.2 Identifying Head Specializations

Previous work pointed out some specific roles played by different heads in the softmax Transformer model (Voita et al., 2018; Tang et al., 2018; Voita et al., 2019). Identifying the specialization of a head can be done by observing the type of tokens or sequences that the head often assigns most of its attention weight; this is facilitated by sparsity.

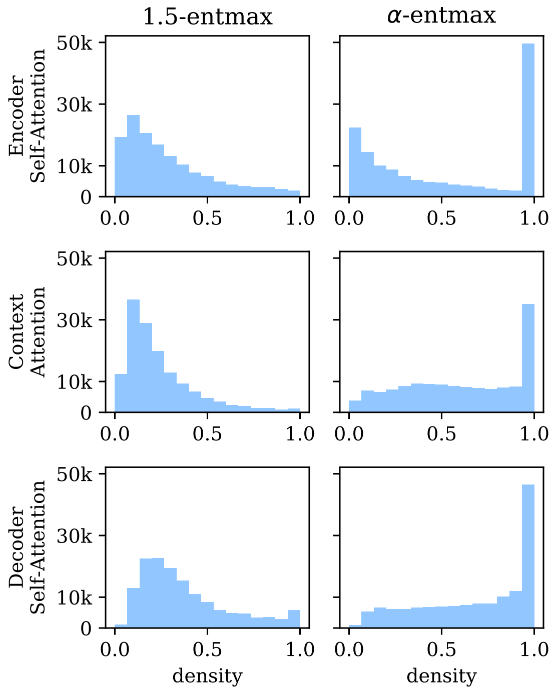

Figure 4: Distribution of attention densities (average number of tokens receiving non-zero attention weight) for all attention heads and all validation sentences. When compared to 1.5-entmax, α-entmax distributes the sparsity in a more uniform manner, with a clear mode at fully dense attentions, corresponding to the heads with low α. In the softmax case, this distribution would lead to a single bar with density 1.

Positional heads. One particular type of head, as noted by Voita et al. (2019), is the positional head. These heads tend to focus their attention on either the previous or next token in the sequence, thus obtaining representations of the neighborhood of the current time step. In Figure 7, we show attention plots for such heads, found for each of the studied models. The sparsity of our models allows these heads to be more confident in their representations, by assigning the whole probability distribution to a single token in the sequence. Concretely, we may measure a positional head's **confidence** as the average attention weight assigned to the previous token. The softmax model has three heads for position -1, with median confidence 93.5%. The 1.5-entmax model also has three heads for this position, with median confidence 94.4%. The adaptive model has four heads, with median confidences 95.9%, the lowest-confidence head being dense with $\alpha = 1.18$, while the highest-confidence head being sparse ($\alpha = 1.91$).

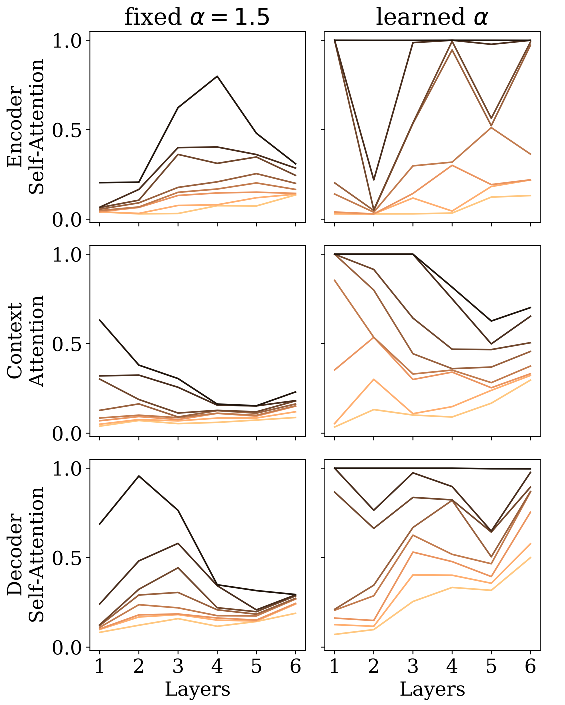

Figure 5: Head density per layer for fixed and learned α. Each line corresponds to an attention head; lower values mean that that attention head is sparser. Learned α has higher variance.

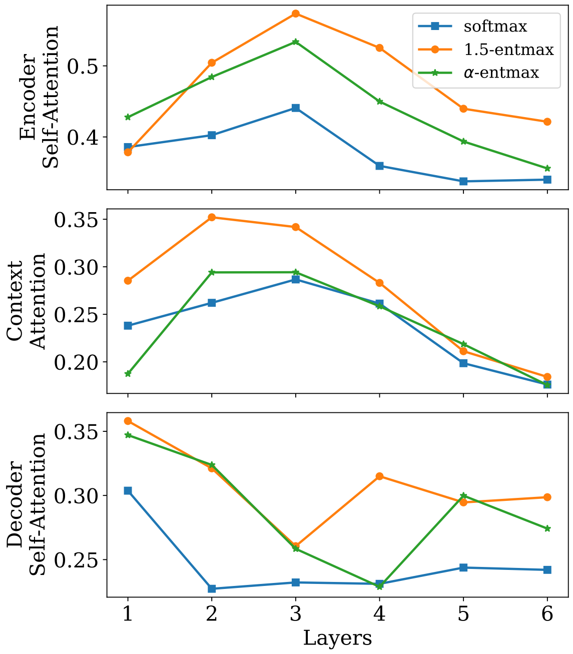

Figure 6: Jensen-Shannon Divergence between heads at each layer. Measures the disagreement between heads: the higher the value, the more the heads are disagreeing with each other in terms of where to attend. Models using sparse entmax have more diverse attention than the softmax baseline.

For position $+1$, the models each dedicate one head, with confidence around 95%, slightly higher for entmax. The adaptive model sets $\alpha = 1.96$ for this head.

BPE-merging head. Due to the sparsity of our models, we are able to identify other head specializations, easily identifying which heads should be further analysed. In Figure 8 we show one such head where the α value is particularly high (in the encoder, layer 1, head 4 depicted in Figure 2). We found that this head most often looks at the current time step with high confidence, making it a positional head with offset 0. However, this head often spreads weight sparsely over 2-3 neighboring tokens, when the tokens are part of the same BPE cluster[2] or hyphenated words. As this head is in the first layer, it provides a useful service to the higher layers by combining information evenly within some BPE clusters.

For each BPE cluster or cluster of hyphenated words, we computed a score between 0 and 1 that corresponds to the maximum attention mass assigned by any token to the rest of the tokens inside the cluster in order to quantify the BPE-merging capabilities of these heads.[3] There are not any attention heads in the softmax model that are able to obtain a score over 80%, while for 1.5-entmax and α-entmax there are two heads in each (83.3% and 85.6% for 1.5-entmax and 88.5% and 89.8% for α-entmax).

Interrogation head. On the other hand, in Figure 9 we show a head for which our adaptively sparse model chose an α close to 1, making it closer to softmax (also shown in *encoder, layer 1, head 3* depicted in Figure 2). We observe that this head assigns a high probability to question marks at the end of the sentence in time steps where the current token is interrogative, thus making it an interrogation-detecting head. We also observe this type of heads in the other models, which we also depict in Figure 9. The average attention weight placed on the question mark when the current token is an interrogative word is 98.5% for softmax, 97.0% for 1.5-entmax, and 99.5% for α-entmax.

Furthermore, we can examine sentences where some tendentially sparse heads become less so, thus identifying sources of ambiguity where the head

[2] BPE-segmented words are denoted by $\sim$ in the figures.

[3] If the cluster has size 1, the score is the weight the token assigns to itself.

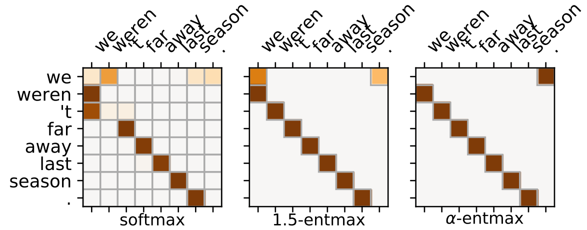

Figure 7: Self-attention from the most confidently previous-position head in each model. The learned parameter in the α-entmax model is $\alpha = 1.91$. Quantitatively more confident, visual inspection confirms that the adaptive head behaves more consistently.

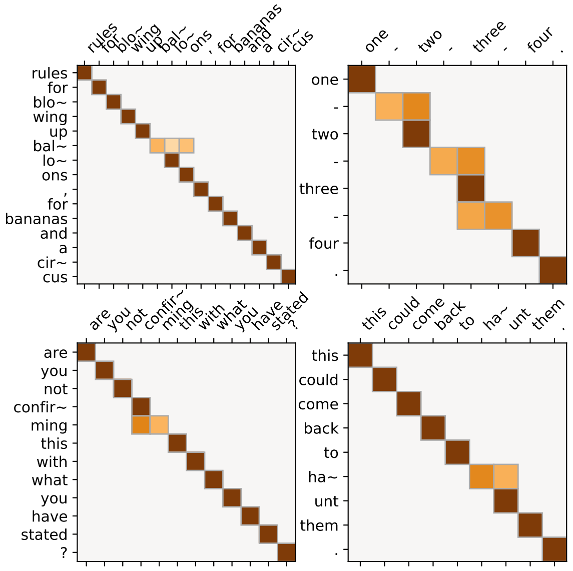

Figure 8: BPE-merging head ($\alpha = 1.91$) discovered in the α-entmax model. Found in the first encoder layer, this head learns to discover some subword units and combine their information, leaving most words intact. It places 99.09% of its probability mass within the same BPE cluster as the current token: more than any head in any other model.

is less confident in its prediction. An example is shown in Figure 10 where sparsity in the same head differs for sentences of similar length.

6 Related Work

Sparse attention. Prior work has developed sparse attention mechanisms, including applications to NMT (Martins and Astudillo, 2016; Malaviya et al., 2018; Niculae and Blondel, 2017; Shao et al., 2019; Maruf et al., 2019). Peters et al. (2019) introduced the entmax function this work builds upon. In their work, there is a single attention mechanism which is controlled by a fixed α. In contrast, this is the first work to allow such attention mappings to *dynamically* adapt their curvature and sparsity, by automatically adjusting the continuous α parameter. We also provide the first results using sparse attention in a Transformer model.

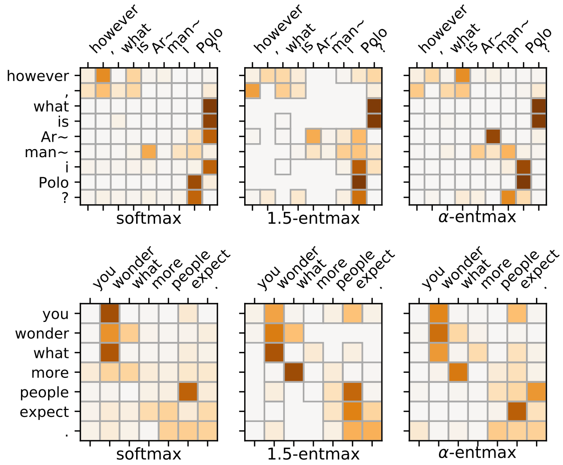

Figure 9: Interrogation-detecting heads in the three models. The top sentence is interrogative while the bottom one is declarative but includes the interrogative word "what". In the top example, these *interrogation heads* assign a high probability to the question mark in the time step of the interrogative word (with $\geq 97.0\%$ probability), while in the bottom example since there is no question mark, the same head does not assign a high probability to the last token in the sentence during the interrogative word time step. Surprisingly, this head prefers a low $\alpha = 1.05$, as can be seen from the dense weights. This allows the head to identify the noun phrase "Armani Polo" better.

Fixed sparsity patterns. Recent research improves the scalability of Transformer-like networks through static, fixed sparsity patterns (Child et al., 2019; Wu et al., 2019). Our adaptively-sparse Transformer can dynamically select a sparsity pattern that finds relevant words regardless of their position (*e.g.*, Figure 9). Moreover, the two strategies could be combined. In a concurrent line of research, Sukhbaatar et al. (2019) propose an adaptive attention span for Transformer language models. While their work has each head learn a different contiguous span of context tokens to attend to, our work finds different sparsity patterns in the same span. Interestingly, some of their findings mirror ours – we found that attention heads in the last layers tend to be denser on average when compared to the ones in the first layers, while their work has found that lower layers tend to have a shorter attention span compared to higher layers.

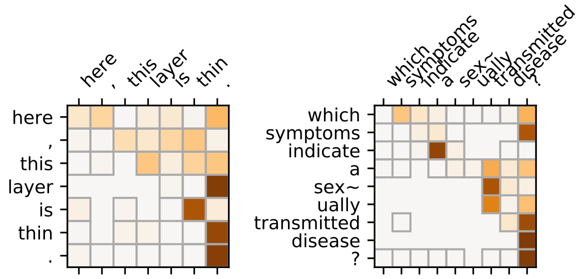

Figure 10: Example of two sentences of similar length where the same head ($\alpha = 1.33$) exhibits different sparsity. The longer phrase in the example on the right "a sexually transmitted disease" is handled with higher confidence, leading to more sparsity.

Transformer interpretability. The original Transformer paper (Vaswani et al., 2017) shows attention visualizations, from which some speculation can be made of the roles the several attention heads have. Mareček and Rosa (2018) study the syntactic abilities of the Transformer self-attention, while Raganato and Tiedemann (2018) extract dependency relations from the attention weights. Tenney et al. (2019) find that the self-attentions in BERT (Devlin et al., 2019) follow a sequence of processes that resembles a classical NLP pipeline. Regarding redundancy of heads, Voita et al. (2019) develop a method that is able to prune heads of the multi-head attention module and make an empirical study of the role that each head has in self-attention (positional, syntactic and rare words). Li et al. (2018) also aim to reduce head redundancy by adding a regularization term to the loss that maximizes head disagreement and obtain improved results. While not considering Transformer attentions, Jain and Wallace (2019) show that traditional attention mechanisms do not necessarily improve interpretability since softmax attention is vulnerable to an adversarial attack leading to wildly different model predictions for the same attention weights. Sparse attention may mitigate these issues; however, our work focuses mostly on a more mechanical aspect of interpretation by analyzing head behavior, rather than on explanations for predictions.

7 Conclusion and Future Work

We contribute a novel strategy for adaptively sparse attention, and, in particular, for adaptively sparse Transformers. We present the first empirical analysis of Transformers with sparse attention mappings (*i.e.*, entmax), showing potential in both translation accuracy as well as in model interpretability.

In particular, we analyzed how the attention heads in the proposed adaptively sparse Transformer can specialize more and with higher confidence. Our adaptivity strategy relies only on gradient-based optimization, side-stepping costly per-head hyper-parameter searches. Further speedups are possible by leveraging more parallelism in the bisection algorithm for computing α-entmax.

Finally, some of the automatically-learned behaviors of our adaptively sparse Transformers – for instance, the near-deterministic positional heads or the subword joining head – may provide new ideas for designing static variations of the Transformer.

Acknowledgments

This work was supported by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contracts UID/EEA/50008/2019 and CMUPERI/TIC/0046/2014 (GoLocal). We are grateful to Ben Peters for the α-entmax code and Erick Fonseca, Marcos Treviso, Pedro Martins, and Tsvetomila Mihaylova for insightful group discussion. We thank Mathieu Blondel for the idea to learn α. We would also like to thank the anonymous reviewers for their helpful feedback.

References

Brandon Amos and J. Zico Kolter. 2017. OptNet: Differentiable optimization as a layer in neural networks. In *Proc. ICML*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Mathieu Blondel, André FT Martins, and Vlad Niculae. 2019. Learning classifiers with Fenchel-Young losses: Generalized entropies, margins, and algorithms. In *Proc. AISTATS*.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proc. Workshop on Statistical Machine Translation*.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proc. WMT*.

M Cettolo, M Federico, L Bentivogli, J Niehues, S Stüker, K Sudoh, K Yoshino, and C Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proc. IWSLT*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse Transformers. *preprint arXiv:1904.10509*.

Frank H Clarke. 1990. *Optimization and Nonsmooth Analysis*. SIAM.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. 2018. Latent alignment and variational attention. In *Proc. NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. ICML*.

Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. 2016. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *preprint arXiv:1607.05447*.

Michael Held, Philip Wolfe, and Harlan P Crowder. 1974. Validation of subgradient optimization. *Mathematical Programming*, 6(1):62–88.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proc. NAACL-HLT*.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proc. WNMT*.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R Lyu, and Tong Zhang. 2018. Multi-Head Attention with Disagreement Regularization. In *Proc. EMNLP*.

Christos Louizos, Max Welling, and Diederik P Kingma. 2018. Learning sparse neural networks through L_0 regularization. *Proc. ICLR*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*.

Chaitanya Malaviya, Pedro Ferreira, and André FT Martins. 2018. Sparse and constrained attention for neural machine translation. In *Proc. ACL*.

David Mareček and Rudolf Rosa. 2018. Extracting syntactic trees from Transformer encoder self-attentions. In *Proc. BlackboxNLP*.

André FT Martins and Ramón Fernandez Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proc. of ICML*.

Sameen Maruf, André FT Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. *preprint arXiv:1903.08788*.

Graham Neubig. 2011. The Kyoto free translation task. `http://www.phontron.com/kftt`.

Vlad Niculae and Mathieu Blondel. 2017. A regularized framework for sparse and structured neural attention. In *Proc. NeurIPS*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proc. WMT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*.

Ben Peters, Vlad Niculae, and André FT Martins. 2019. Sparse sequence-to-sequence models. In *Proc. ACL*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *preprint*.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in Transformer-based machine translation. In *Proc. BlackboxNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. ACL*.

Wenqi Shao, Tianjian Meng, Jingyu Li, Ruimao Zhang, Yudian Li, Xiaogang Wang, and Ping Luo. 2019. SSN: Learning sparse switchable normalization via SparsestMax. In *Proc. CVPR*.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive Attention Span in Transformers. In *Proc. ACL*.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? A targeted evaluation of neural machine translation architectures. In *Proc. EMNLP*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proc. ACL*.

Constantino Tsallis. 1988. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52:479–487.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NeurIPS*.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proc. ACL*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proc. ACL*.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *Proc. ICLR*.