大语言模型节省一半算力的最简单方法：只用中文为唯一推理语言

1 保守方案:加一个语言按钮。区分英文和中文。

语言模型的结构和算力可以分为三块:输入层、计算层、输出层。AI 要输出一个字，必须把词表里面所有的字都计算一次。所以词表越大，消耗的算力就越多。以现在为例，词表有 15 万个字，10 万个英文单词，5 万个中文。在中文对话中，AI 每次都要去算那 10 万的英文单词，输出层 2/3 的算力都是浪费的。解决这种浪费的方法很简单:就是中文问答在生成答案时，只从中文词表中取词。加一个语言按钮就能省掉 2/3 的输出层算力。

2 激进方案:AI 只使用单一语言，其他语言交给翻译器。AI 只使用一种语言，那么 AI 的词表也只需要一种语言就够了。这除了减少输出层的算力外，还会带来其他好处。

2.1 AI 只使用英文为唯一语言。

英文单词和单词之间有一个空格，词边界清晰。英文的一个单词就是中文的一个词组，一个单词等于两个中文字。但中文的字和词中间没有空格，词边界不清晰，导致中文存在分词困境，只能使用单字作为 token，不能使用词组，后果是，同样一句话，中文的 token 数量是英文的两倍，计算层要消耗两倍的算力。如果 AI 只以英文为语言，则中文的分词困境就不存在了。中文被翻译成英文送入计算层，计算层消耗的算力直接减半。当然，中文的语言丰富性将荡然无存。

2.2 AI 只使用中文为唯一语言更好。

2.2.1 中文分词难题的解决方案:用输入法取代分词器。

中文输入法早已解决中文的词边界问题。我们现在却把输入法的分词信息去掉，再让分词器去分词。结果分词器就是个废物，根本无法正确分词。所以至今中文只能 1 字 1token。分词器消耗了输入层大部分的算力，依然只是查字典。因为英文必须有分词器，所以如果只用中文，分词器就可以丢掉了，只用查字典程序，就能节省输入层的大部分算力。

英文单词之间有空格，只是因为输入法在输出英文时自动添加了空格。我们也可以让输入法在输出中国的字和词时自动添加隐形空格。中文的分词问题就解决了。中文在脱离分词困境以后，就可以转而使用大量的词组，计算时的 token 数量可以成倍减少。计算层可以轻松节省一半的算力。

2.2.2 词边界确定后，中文可以大量使用语义融合技术。把两个字的语义临时融合成一个 token 语义，送入计算层。也就是说中文其实只需要 1 字 1token 就可以了，根本不需要添加十万词组。词组是可以通过语义融合生成的。我猜一个词组的所有语义信息其实已经分散在了这两个字的向量里。一旦两个字进行语义融合计算，就会自然涌现这个词的含义。现在没有使用这个技术是因为中文没有解决词边界问题，而英文根本用不了这个技术。所以中文的 Token 词表数量可以大幅减少到等于字的数量。

2.2.3 中文还可以把非常用的生僻字，踢出词表，用□代替。也就是说中文的词表只需要 4000 个字就够了。输出层的算力直降 90%以上。

2.2.4 中文还有一个核心优势，中文的高效在算力上的表现为词组的语义清晰。英文几乎所有的单词都是一词多义的。中文字是多义的，但词的意思却很集中。比如，行字，有行业行动等意思。但组合成行业和行动这个词组后，则语义单一。相反，英文的银行这个单词，同时又有河岸等好几种意思。每个 token 的向量维度，英文所需要的维度比中文更多，英文所需的算力天然比中文要多。

3 符号的算力浪费

比如千亿模型，每个符号的向量有 1 万多个。符号本身承载的信息是极少的。根本不需要这么多的向量。我们能不能就是把符号单独做个表。把这个表的向量数量降为 500。然后对计算层进行套壳。符号向量在进入计算存钱，自动用零补足不足的向量。跟计算层对齐。我曾经用千问实验过这个方案，可能是可行的。但前提可能在于千问采用了 mrl 技术。