1 三种不同的属性推导方式的比较

2 属性推导时的参数分析

2.1 关于的大小

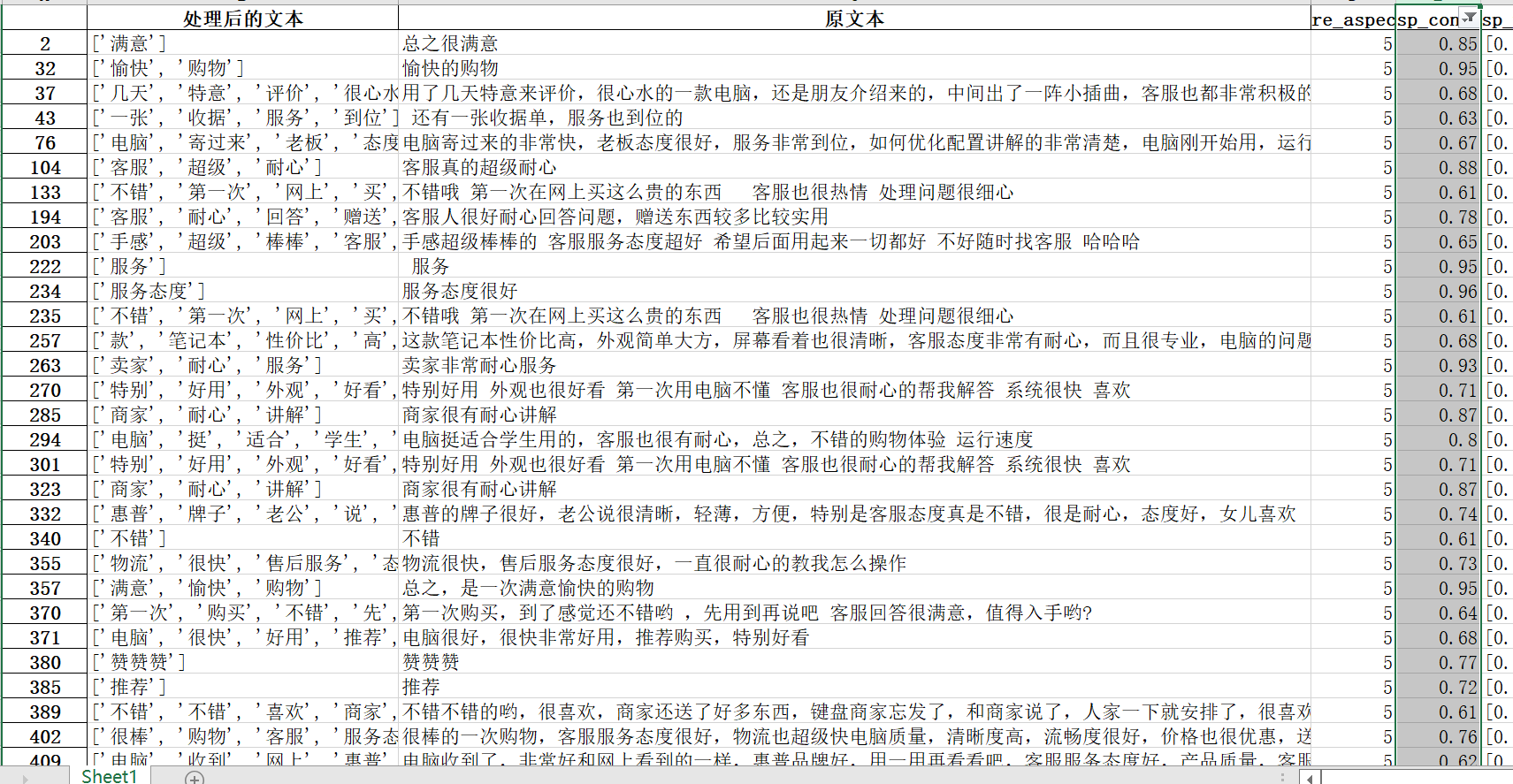


图 k=10，时，category=5的句子分类结果



图 k=10，时，category=5的句子分类结果

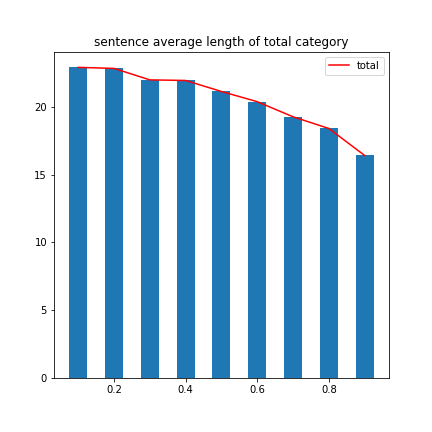


图 k=10时，句子的平均长度关于的变化

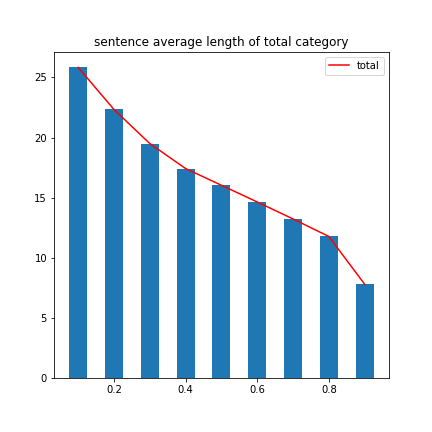


图 k=30时，句子的平均长度关于的变化

句子的平均长度随着的增大而减小，这就意味着，的增大一定程度上剔除了一些信息。如果想要获得更加精确的属性，我们可以选取较大的用于属性推导；如果想要看到更多的信息，我们可以选取较小的用于属性推导

另外，这一现象在k值较大时更加明显。一个可能的解释是，比较大的k会迫使属性的划分更加细化，这时的一个属性可能需要更短的句子来解释。

2.2 关于k的大小对推导结果的影响



图 k=10，时，category=5的句子分类结果



图 k=30，时，category=21的句子分类结果

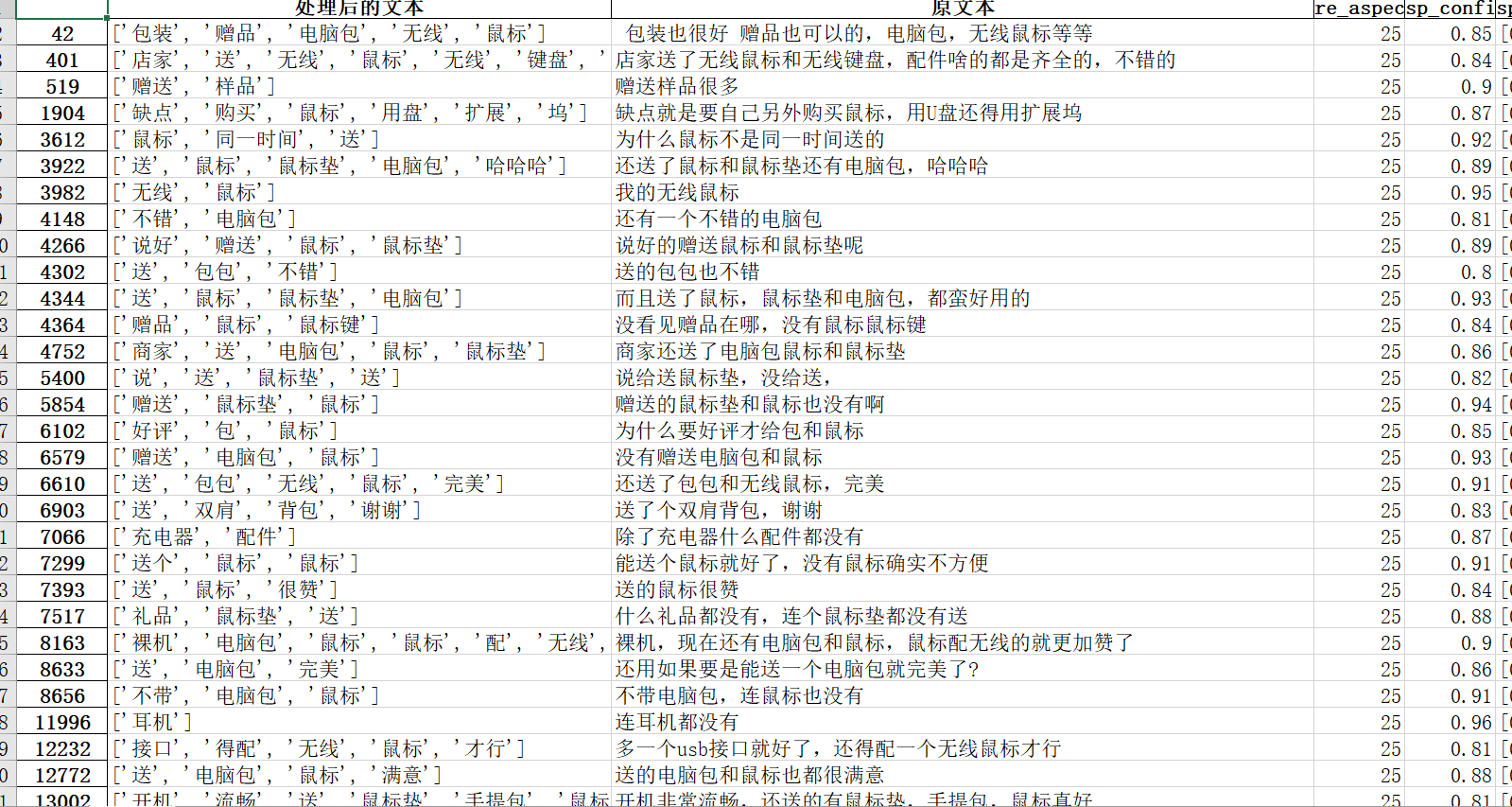


图 k=30，时，category=25的句子分类结果

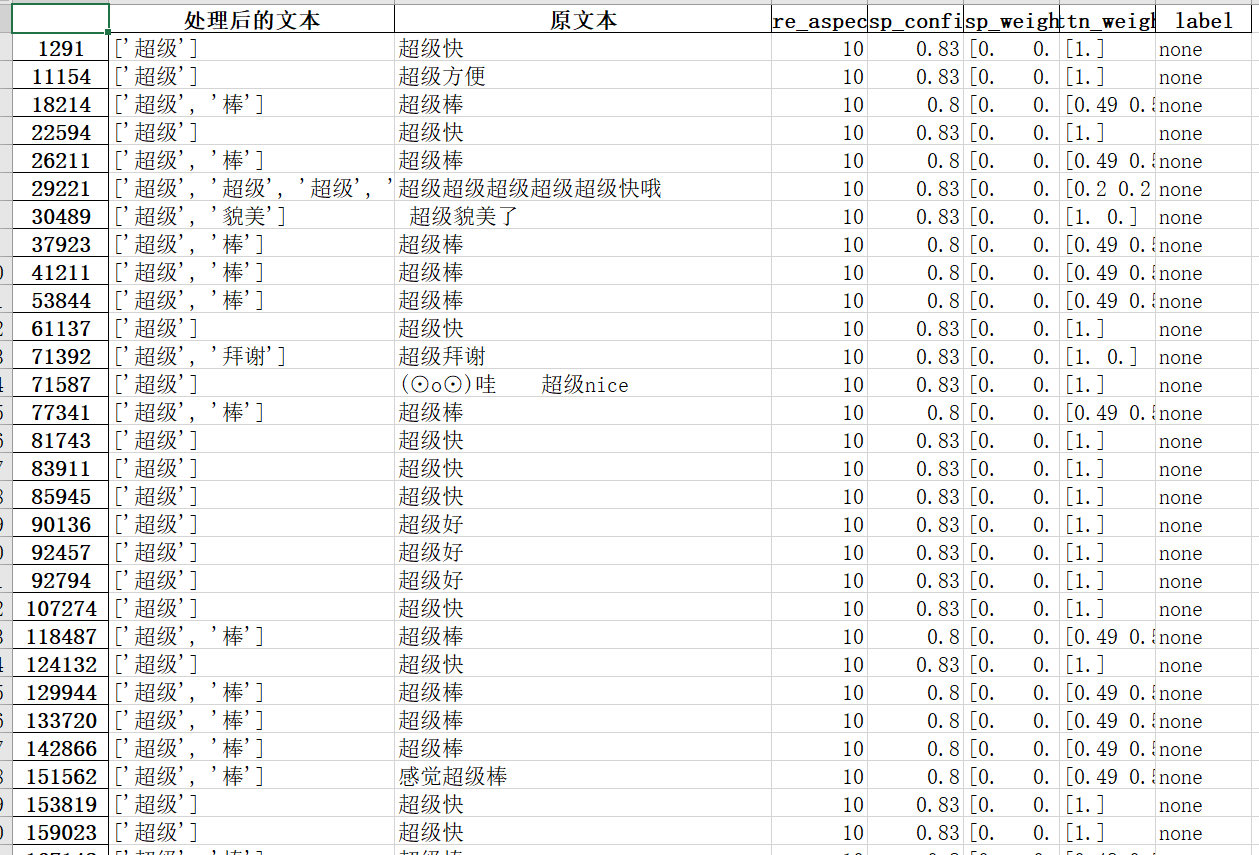


图 k=30，时，category=10的句子分类结果

k=10时，在我们的数据中，虽然每个类别的含义是很不同的。但是，每个属性可能存在着多重的含义，这时的句子普遍偏长；

k=30时，属性的划分更加细化了。如图所示，k=10时的第5个类别，在k=30时可能被细化为了3个主要的类别，包括：客服服务、赠品和一个非属性“赞”。

k值越大，越容易出现细化的属性，容易找到更多的属性，但同时也会找到更多的非属性。同时，每个类别下的句子数量也会减少。

3 消融实验

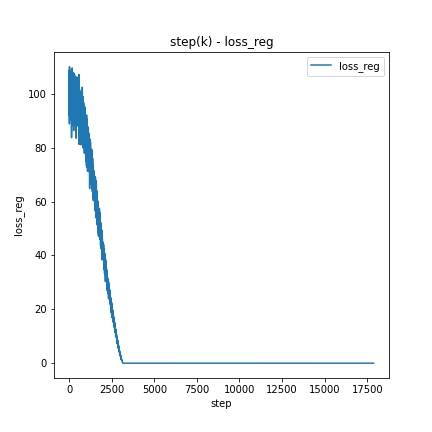


图 k=30，正常情况下正交正则化损失变化

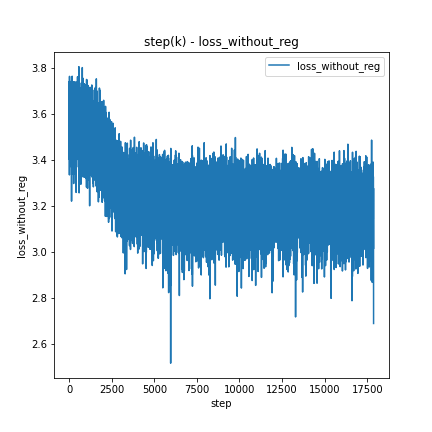


图 k=30，正常情况下负采样损失变化

正常情况下正交正则化损失在3000步左右收敛，此后都维持在一个很低的水平（0左右）；负采样损失在4000步左右收敛，此后在3.0-3.4之间波动。

3.1 不用k-means初始化aspect embedding后会怎样

3.2 不用正交正则化损失会怎样

3.2.1 从损失上看

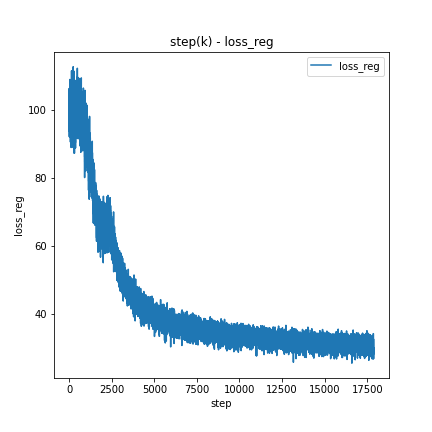


图 k=30，正交正则化损失的变化

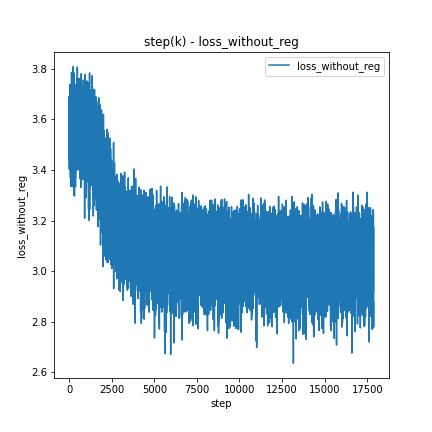


图 k=30，负采样损失的变化

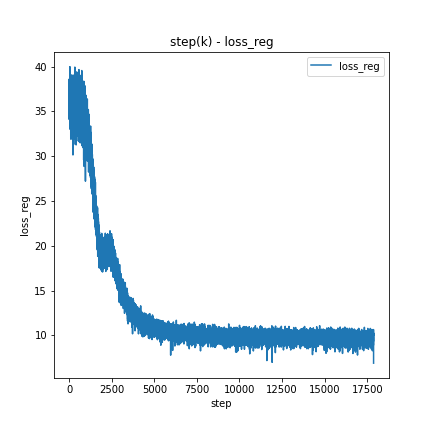


图 k=10，正交正则化损失的变化

从损失上看，

虽然正交正则化损失没有参与模型的训练，但是它依然在训练过程中缓慢地收敛到了一个不是很理想的数值（k=30时这个数值在30-32之间波动）。而收敛速度与k值的大小有关，k值越大，它收敛就越慢；而另一个损失——负采样损失的收敛速度基本与原模型一致，甚至收敛到了一个更低的水平（2.8-3.2）

从结果上看，

它的aspect embedding也有着一定的属性推导作用。但是，在后期的句子分类实验中，我们发现aspect embedding所对应的self attention并不能很好地给aspect embedding分配权重，大多数情况下，self attention会给aspect embedding分配比较平均的权重，在我们的实验中，几乎没有出现confidence大于0.5的句子。而在句子分类过程中，虽然每个类别下也有一定的主题，但是存在着许多的噪声，也就是说，很容易把一些无关的类别分到一起。并且，加大k的值并不能很好地缓解这个问题。

我们由此猜测，正交正则化的损失的存在，将会促进aspect embedding所对应的self attention给aspect embedding分配一个较大的权重。也就是，aspect embedding的不同，将会促进self attention分配一个较极端的权重。