Entity Linking and Discovery via Arborescence-based Supervised Clustering – May 2022

Arxiv version available only, might not be Published

Continuing work on clustering-based inference

Introduction /Idea –

Previous model following this clustering approach to perform linking used cross encoder to train models, limitations for which was it picked mention candidates from within the document for training , But here Bi Encoder (transformer based ) leading to minor downfall in accuracy but major improvement in training time.

Importance of using this approach

Also, improvement in entity linking task and discovery is obtained

Also, this model performs the task of entity discovery and zero shot effectively

Zero Shot vs Entity Discovery 🡪

Zero shot 🡪 learning is when entities in KB that do not have any labelled training data.

Linking + discovery 🡪 Complete KB of entities may not be known in advance and new entities must be discovered. By removing a fraction of entities from the KB and demonstrate that our training and inference procedures are better suited for this task. Clusters represent a new entity if set of mention is clustered wo an entity node

Steps of method

1. Graph based dissimilarity measure
2. Models to provide edge weight
3. Approach for inferring edge weights
4. Building constrained clusters

Now going stepwise

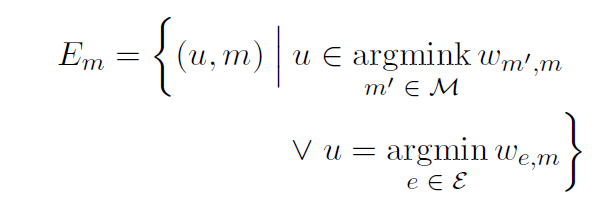
🡪Firstly, Dissimilarity Score is defined as minimum amongst highest weight edges in paths between two nodes in graph.

🡪Edge weights will be calculated using Affinity models. Which are implemented using Bi Encoder transformers.

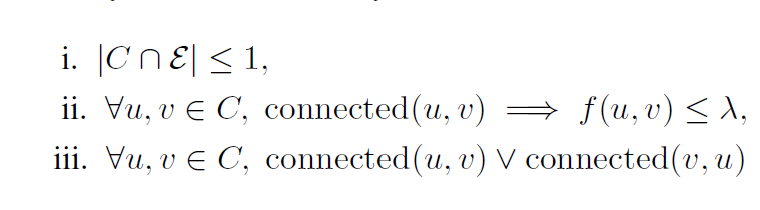
Graph construction 🡪

It optimizes the models based on what nodes are connected and what path exists in the graph.

It constructs edge set Em for m in M following , first selecting k nearest neighbours and then edges are picked following the convention that



Clustering Step 🡪

is the process of partition

Here C(u) represent possible cluster with entity as u.

So, satisfying these constraint, following algorithm is used:

1.Remove edge with score less than lambda. And arrange edges in descending order.

2.Pick an edge (u,v) , drop it from the graph , Check If v can still be reached by some entity in the graph permanently drop (u,v) else retain it .

So ,there is a directed path from the entity node to every other node in the cluster. This structure is often referred to as the minimum spanning arborescence, thus lending its name to our method,

So,

This model operates on all mentions in the corpus instead of within document setting

Purpose of using this inductive bias 🡪 Instead of using sum of weights in a path gives an inductive bias for even weakly connected mentions is suitable for coreference.

Here affinity score is calculated simply multiplying associated encoded representation.

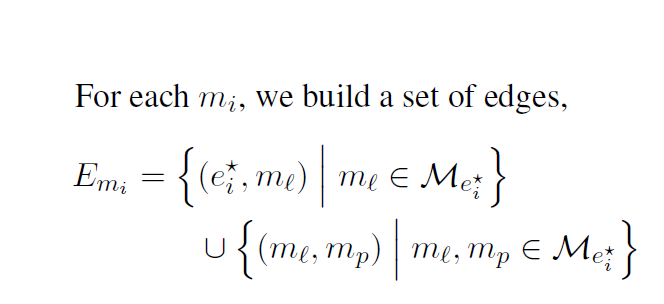
Now since we know the score between mentions we can calculate weight between non connected mentions .

Model 🡪

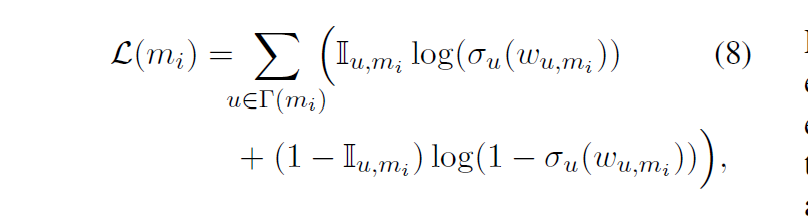
Checks wheather there is some other entity which can reach the node with weight highest weight will be removed. Graph is constructed only once

Training 🡪

The resultant edges ensure that each connected component contains exactly one entity(namely, the ground truth entity for the mentions in that component). We then sparsify by computinga partitioned target graph

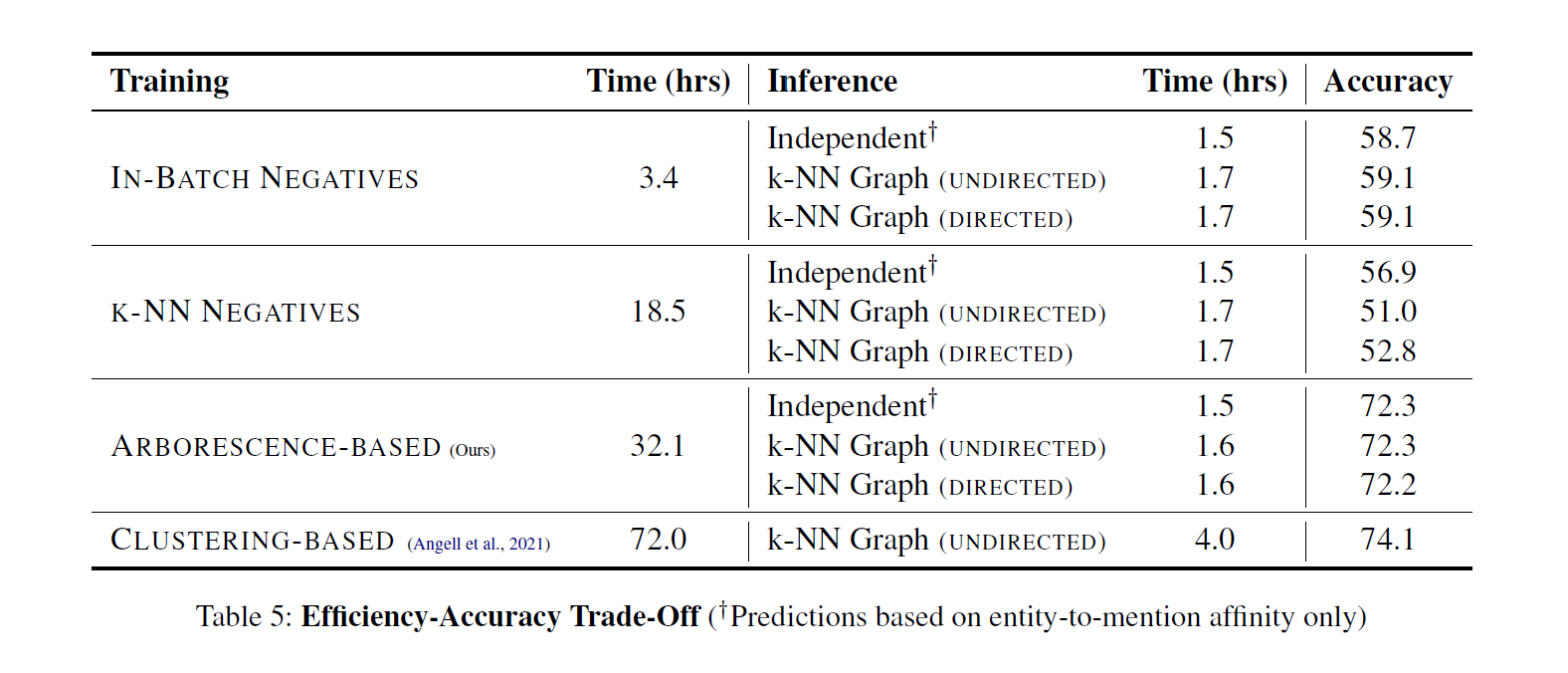


Loss function:



Optimizing this loss function requires simultaneously increasing the likelihood of the positive edges and decreasing the likelihood of the negative edges as in the definition.

Results:



Training time is reduced significantly with accuracy tradeoff

Will update this review as things get more clear.