Open Domain Information Extraction with Harmonic Energy Functions

Chirag Nagpal[1], Kyle Miller and Artur Dubrawski[2]

[1]Language Technologies Institute, [2]Auton Lab, Robotics Institute, Carnegie Mellon University
{ chiragn, mille856, awd} @ andrew.cmu.edu

Objectives

We build upon the Harmonic Energy Functions used previously in an Active Learning Framework

- Named Entities extracted with Distant Supervision from external Knowledge Bases are exploited to describe a Graph with weighted edges.
- Entities to provide context, and constrain search to utilise Active Search for Information Extraction from Open Domain data.
- Scalabilty issues arising due to large number of nodes are addressed with numerical techniques, that speed solution of Linear Systems.

Introduction

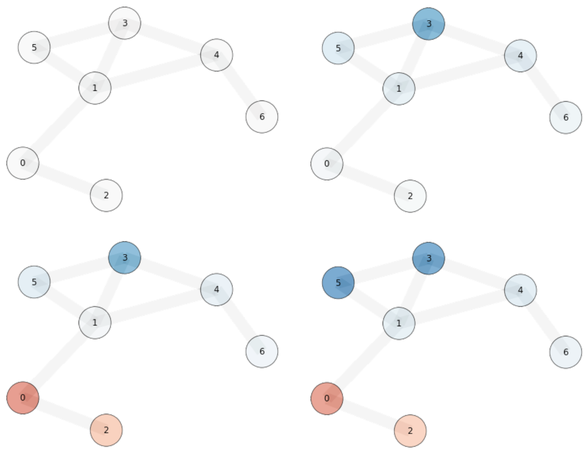


Figure 1: In each iteration, the a label associated with the node is queried. With each new label, the activations change, so as to minimise the Energy over the Graph.

$$\arg\min_f E(f) = \sum_{i \in L}(y_i - f_i)^2 D_{ii} + \lambda(w_0 \sum_{i \in U}(f_i - \pi)^2 D_{ii} + \sum_{i,j}(f_i - f_j)^2 A_{ij})$$

where, $A_{ij} = K(x_i, x_j), D_{ii} = \sum_j K(x_i, x_j)$ and λ, w_0 are regularising constants

- Active Search [1], minimises the Energy, $E(f)$.
- f is the 'activations' of each node in the Graph.
- Instead of a Kernel, $K(\cdot, \cdot)$ as edge, entities are exploited as nodes with documents. This results in bi-partite graphs over which energy minimisation is performed.

Architecture

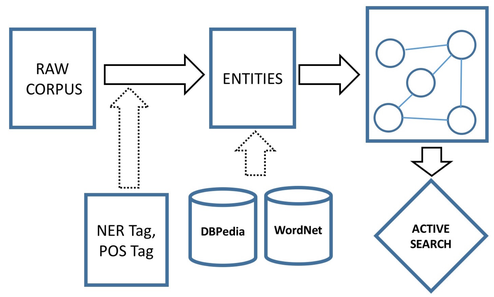


Figure 2: Data Preprocessing Pipeline

To extract entities, we exploit external KBs:

- Wordnet: Named Entity Recogniser and Part of Speech Tagger trained on the Penn Treebank Corpus is employed along with Lesk, to perform Word Sense Disambiguation.
- DBPedia: 'Spotlight'[2] an open-source tool to map text to Entities.

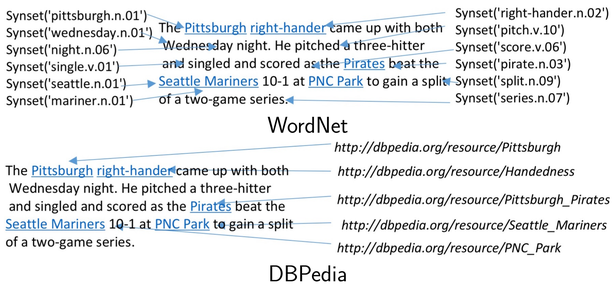


Figure 3: Comparative Entities Extracted with WordNet and DBPedia on a representative sample.

- Active Search, is applied to 20 Newsgroups for a One-Vs-Rest Classification task.
- AS on Bi-Partite graphs with Entities significantly outperforms AS on Graphs with Cosine Distance between nodes.

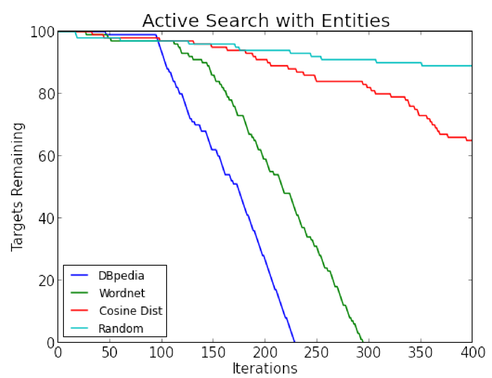
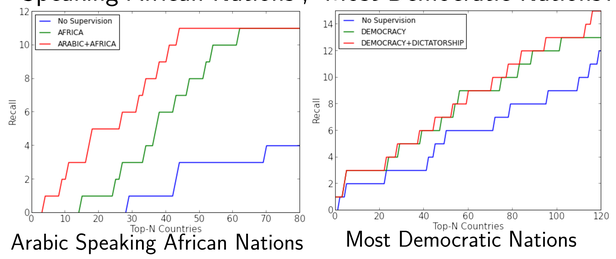


Figure 4: Harmonic Energy Minimisation - Active Search

Information Extraction

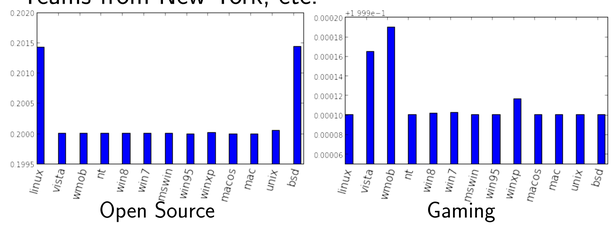
- Energy Minimisation is extended towards a more Generic Open Domain Information Extraction System
- A structured dataset curated from Wikipedia articles of countries is utilised to answer queries such as 'Arabic Speaking African Nations', 'Most Democratic Nations'.



Arabic Speaking African Nations | Most Democratic Nations

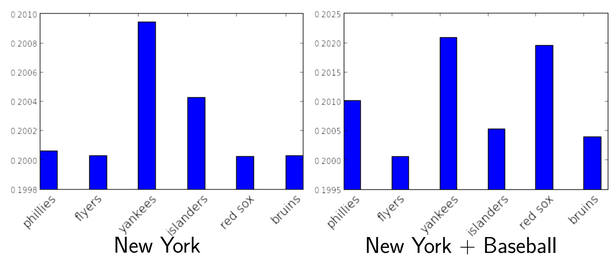
Figure 5: Graph extracted from WordNet on Wikipedia pages of 250 countries. We compare recall by labelling certain 'interesting' entities as Positive.

- Robustness is tested on the more unstructured dataset, 20 Newsgroups, to answer simple queries like the Open Source or Gaming Operatin Systems, Sports Teams from New York, etc.



Open Source | Gaming

Figure 6: Activations of Entitites representing 'Operating Systems', on a Graph extracted from DBPedia on 20 Newsgroups.



New York | New York + Baseball

Figure 7: Activations of Entities Representing 'Sports Teams' on a Graph extracted from DBPedia on 20 Newsgroups.

Future Work

- Treat change in $f^{(t+1)}$ w.r.t. $f^{(t)}$ as Impact Factor (IF).
- Faster computation given IF is a function of f or $f^\intercal.f$
- Investigate use of a classifier, to learn IF.

Scalability

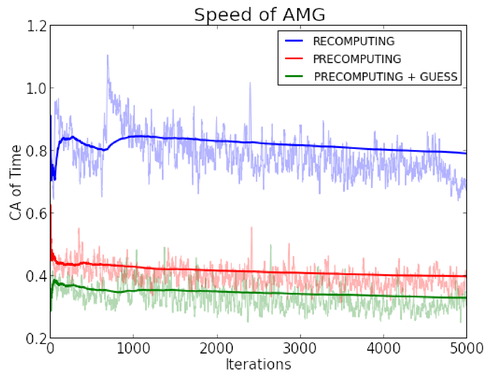


Figure 8: Scalability Improvements with Alegebric Multigrid

$$f = (\tilde{D} + \lambda(D - A))^{-1}\tilde{D}y'$$

where, $\tilde{D} = \begin{bmatrix} D_L & 0 \\ 0 & \lambda w_0 D_U \end{bmatrix}$

As, E is convex in f, we can solve the given Linear System, using Algebric Multigrid[3], which is further speeded, by precomputing the Restriction and Prolongation matrix, and reusing them in each iteration.

Algorithm 1: Search with Precomputed Grids

Input : w_0, η, π

Initialise Graph, G with the targets

$t \leftarrow 0$;
if $t == 0$ **then**
  $A, y \leftarrow$ setData(G);
  $R, P \leftarrow$ ComputeGrids(A);
  $f^{(t+1)} \leftarrow$solveAMG(R, P, A, y);
else
  $A, y \leftarrow$ setData(G);
  $f^{(t+1)} \leftarrow$solveAMG$(R, P, A, y, f^{(t)})$;
end

References

[1] Xuezhi Wang, Roman Garnett, and Jeff Schneider.
Active search on graphs.
In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 731–738. ACM, 2013.

[2] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes.
Improving efficiency and accuracy in multilingual entity extraction.
In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

[3] W. N. Bell, L. N. Olson, and J. B. Schroder.
PyAMG: Algebraic multigrid solvers in Python v3.0, 2015.
Release 3.0.

Auton Lab | Language Technologies Institute | The Robotics Institute
Carnegie Mellon University