Named entity linking with graphical models and deep learning

Survey and exploration

Sammy Khalife

Ecole Polytechnique, LIX, Dascim

khalife@lix.polytechnique.fr

Abstract

The task of named entity discovery (NED) in machine learning and natural language processing means the ability for a program to extract some pre-defined sets of words in a vocabulary (called named entities : names, places, locations, ...), and then to identity them by linking to a pre-existing database. The first subtask is called named entity recognition (NER) and is not trivial since we don't have an exhaustive list of this named entities, moreover their text representation can change (For example B. Obama instead of Barack Obama). The second task is named entity linking (NEL).

In this section we will focus on NEL and review main their main existing algorithms : the first category gathers graphical models (including graphs and PGMs, and the second one), and the second one deep learning approaches. To do so, we will propose to see how relevant these algorithms can be for this task, compare their ... and propose new algorithms for named entity linking

Objectives

- Provide a survey of named entity linking
- Discuss algorithms pertinence, and computational complexity
- Compare performance on real datasets provided by NIST for named entity discovery (TAC-KBP Challenge)

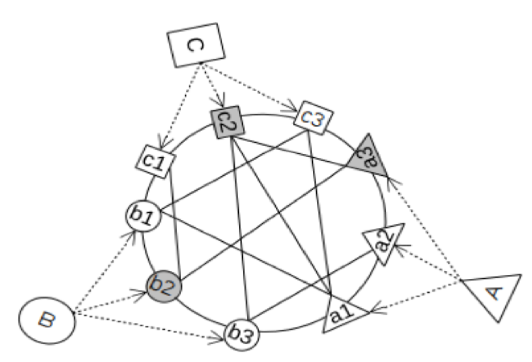
Survey of existing methods

Name entity linking implies having a database at hand. In section Graphs for NED and Graphical models, we are in a **Collective disambiguation context** which set the correspondances jointly with all mentions and entities, then takes into account the links between entities in the database itself.

Graphs for NED

Dense subgraph extraction [5], [2]

Dream : Using the weighted graph built between mentions and entities with local similarities, identify a dense subgraph that contains exactly one mention-entity edge for each mention, yielding the most likely disambiguation



- **Issue :** In general, *NP-hard* with respect to graph size
- [5] and [2] proposes a discarding algorithm (tabou search) based on local similarities with polynomial complexity

Hubs and authorities : HITS algorithm [8]

- [8] Build the weighted graph G of all mentions and entities built with local and global similarities, and run the HITS algorithm (similar to PageRank) on the graph. Hubs and authorites provide the entity linking
- Complexity : polynomial (to be detailed, cf [8])

Graphical models

Modified HMMs for NED [1]

Issue : number of possible state per entity is huge a priori (unlike character recognition where $|E| = |E_i| = 26$, and Viterbi complexity is $O(N \times |S|^2)$ where S number of states and N number of observations

Solution :

- A first step proposed by [1] is to establish a **reduced set of candidates** per mention : $m_i \in E_i$ using mention context
- Second step to use message passing (Viterbi algorithm) to find the most probable Named entity sequence
- HMMs works surprisingly well considering 2 supplementary assumptions
  - context of a mention is a Wikipedia article subsection
  - State sequence with higher probability over 2 orders for Viterbi algorithm: first by natural appearance, and ordered by size of the set of named entities candidates.

Probabilistic bag of Hyperlinks [4]

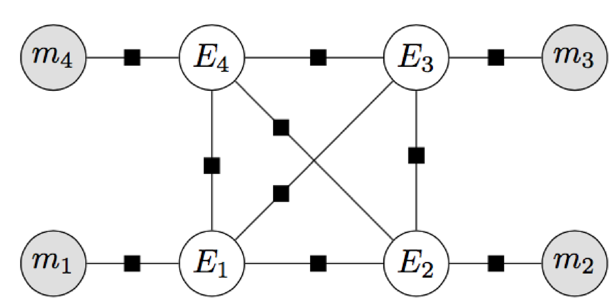
- Probabilistic graphical model with Factor graph
- Probability density within a family of function using maximum entropy principle
- **Bethe Approximation** to decrease inference computational cost
- Message passing to compute marginal probabilities
- Complexity : $O(n^2 \times r^2)$ where r is the number of average entity candidates per mention

Algorithm 1 PBOH [4]

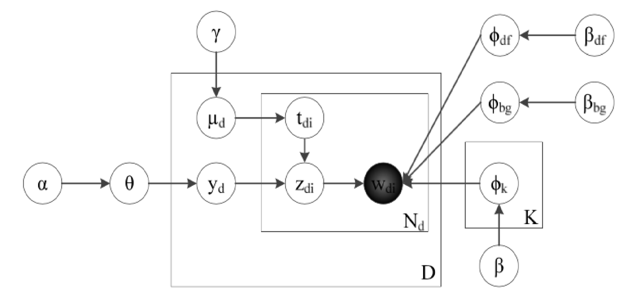
Input : Knowledge Base, Documents with entity mentions, and entities with context
Output : Mentions disambiguated
1 - Establish PGM model with context, and Bethe approximation (Factor graph simplified as a tree)
2 - Graph Inference on parameters using stochastic gradient descent
3 - Compute marginal probabilities using message passing
4 - Keep most probable entity for each mention

Evidence mining for NED [6]

- Iterative inference on probabilistical grapical model to determine underlying entity
- Iterative completion information using external database (such as Wikipedia)
- Differences with LDA
  - Multiple dirichlet priors for topic distibutions over words
  - A document is associated to an unique topic
  - Word label set to different labels (regular, background) from document (defaut, regular, background)
  - Complexity : $O(M \times MaxIter \times Complexity(LDA))$



PBOH without context words



Incremental inference for evidence mining

Figure 1: Two probabilistic graphical models for named entity linking

Deep architectures

Embeddings [9]

- Word embeddings are practical and used for deep learning architectures
- They can be learnt either outside of a deep learning architecture or inside one, as a first layer
- An disambiguation example using pre-trained embeddings, then averaging and ranking has been constructed [9] with complexity $O(m \times e^2)$ where m is the number of mentions and e the number of entities

Convolutional network [7]

- Uses pre-trained mention and context embeddings, entity word and class embeddings
- Cosine similarity is computed at the output of the network between mention and entity embeddings
- Corrupted loss is chosen in a supervised context to train the network (false links and good links) with polynomial complexity

Recurrent batch normalized LSTM for entity linking

- LSTMs have found good results for natural language modeling. Each observation is a mention and its associated representation (context embedding) and the output is the associated entity
- As an application of recurrent batch normalized version [3] with polynomial complexity, I implemented using Tensorflow this network.

Results

There are multiple datasets to compare performance in terms of precision and recall of these algorithms : AIDA A-B, MSNBC, AQUAINT, ACE2004, WNED, and NIST TAC-KBP Challenge of Entity Discovery and Linking (2009 to 2016). Today, there is no unified comparison between probabilistic graphical methods and deep learning methods.

However, **Gerbil** platform provides comparison between graphs methods, and reveals PBOH [4] to be the most accurate of the graphs methods. To the best of my Knowledge, NIST TAC-KBP EDL challenge of 2009-2010 best performance was hold by Convolutional networks using pre-trained embeddings [7]. Comparison with recurrent batch normalized LSTM is being established.

Propositions and current work

Conclusions

We provided a survey of existing algorithms for named entity linking, and propose an adaptation of a recent deep architecture that could capture more information on previous mention and their context. However, the full comparison with other algorithms on real datasets has to be done jointly with PGMs. Hybrid approaches are being investigated. Since NEL is a subtask of NED, one could investigate a joined framework using deep architectures for both NER and NEL. Moreover, hybrid approaches with graphical model and deep architectures are being investigated.

References

[1] Ayman Alhelbawy and Robert J. Gaizauskas. Named entity disambiguation using hmms. In *2013 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Atlanta, Georgia, USA, 17-20 November 2013, Workshop Proceedings*, pages 159–162, 2013.

[2] Ayman Alhelbawy and Robert J Gaizauskas. Graph ranking for collective named entity disambiguation. In *ACL (2)*, pages 75–80, 2014.

[3] Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville. Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*, 2016.

[4] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, pages 927–938. International World Wide Web Conferences Steering Committee, 2016.

[5] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[6] Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1070–1078, New York, NY, USA, 2013. ACM.

[7] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Modeling mention, context and entity with neural networks for entity disambiguation. In *IJCAI*, pages 1333–1339, 2015.

[8] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. Agdistis-graph-based disambiguation of named entities using linked data. In *International Semantic Web Conference*, pages 457–471. Springer, 2014.

[9] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *CoRR*, abs/1601.01343, 2016.

[10] Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. Aida: An online tool for accurate disambiguation of named entities in text and tables. *Proceedings of the VLDB Endowment*, 4(12):1450–1453, 2011.

Acknowledgements

I would like to thank Octavian Ganea from ETH Zurich for having provided detailed explainations on the PBOH implementation, LIX Dascim team members for our exchanges, and BNP Paribas CIB Analytics Consulting for funding this Phd.