Combining Truth Discovery and Open Information Extraction with Active Ensembling

Mouhamadou Lamine BA
Qatar Computing Research Institute
Tornado Tower, West Bay
Doha, Qatar
mlba@qf.org.qa

Laure Berti-Equille
Qatar Computing Research Institute
Tornado Tower, West Bay
Doha ; Qatar
lberti.qf.org.qa

ABSTRACT

Web search engines or open information extraction systems usually reply to users' queries with a set of candidate answers that are often conflicting because they are claimed by multiple information sources. In this context, estimating information veracity is difficult for the users especially when they have no prior knowledge about the trustworthiness of the sources. In this demonstration paper, we showcase a system that supports event/entity and relation extraction based on keyword-search from the Web, processes the conflicting outputs, combines multiple truth finding algorithms with active learning to provide the most likely true answers and determine the most trustworthy sources.

1. INTRODUCTION

[**Lamine:** **Page allocation**]

– 1.25 pages –> abstract + introduction
– 1.5 pages –> Open information extraction + Active Ensembling for Truth Discovery
– 1 pages –> Demonstration System + Scenario
– 0.25 pages –> References

2. OPEN INFORMATION EXTRACTION

- décrire le type d'information auquel on s'intéresse par exemple "factoid claim" - decrire le systeme sur lequel on se base - décrire comment on transforme l'output de OpenIE - donner qq exemples

We rely in this study on open information Web extractors in order to candidates tuples related to "factoid claims" about real-world topics from multiple conflicting Web sources. Open information extractors are unsupervised extraction system that makes a single, data-driven pass over an entire corpus (unstructured texts) and extract a large set of relational tuples without any human intervention.

The input t of the extraction system is a triplet $(e_i, r_{i,j}, e_j)$ where e_i and e_j are called *entities* and $r_{i,j}$ is a *relation*. The semantics of $r_{i,j}$ is that that of a certain relationship between the two given entities. Given this input, the extractor searches into a corpus, consisting of several thousand of sentences, the collections of relational tuples that might satisfy the searched relation. In other terms, the extractor outputs a set of tuples $<(e_i, r_{i,j}, e_j)>$.

Factoid claims. a piece of unverified or inaccurate information that is presented in the present as factual, often as part of a publicity effort, and that is then accepted as true because of frequent repetition.

Extraction of claims. We use TextRunner for the extraction of the set of candidates claims that might correspond to the user query.

3. ACTIVE ENSEMBLING FOR TRUTH FINDING

3.1 Ensembling

- donner idée générale pour introduire ce qu'est l'ensembling - on a besoin de le faire dans le contexte de truth discovery car aucune methode ne bat toutes les autres dans tous les cas de figure - donc on combine les methodes : il y plusieurs façon de combiner par ex. consensus de méthodes, etc. - expliquer quelles méthodes on combine avec leurs avantages et inconvénients

An ensemble is a supervised learning algorithm in the sense that is can be trained and then used to make predictions. An ensembling, or commonly an ensemble-based active learning, is a learning process selecting one classifier type, or appropriate combinaisons of multiple classifier types, to construct ensembles for a given tasks.

3.2 Active Learning Process

- notre approche que l'on défend ici dans la démo est semi supervisée en impliquant de l'utiliseur de façon active en lui demandant s'il peut confirmer des faits (facts) - si on a une ground truth partielle on la "rejoue" cas par cas

4. OUR DEMONSTRATION SYSTEM

4.1 System Architecture and User Interface

The architecture of our demonstration system, given in Figure 1, comprises the following three main components.

User I/O Interface. It represents the main entry point of our application for user interaction. The user I/O interface is composed by a text search area where a given user can enter its search keywords, in terms of a relation, The final result of the overall process will be also show to the users through this component. Finally, the user gives it feebacks via the user I/O interface through the button options or the form.

Information extraction module. This is the information extraction module which considers the input of the user and browsers several Web sources in order to returns the relevant answers. In our system, we rely on TextRunner in order to extract information from Web corpus.

Truth Finding Engine. It corresponds to AllegatorTrack which contains twelve truth finding algorithms with different accuracy according to the types of claims and the characteristics of sources.

Learning Module. We have also a learning method that uses our knwoledge bases of users feedbacks. It enables to learn about the best truth finding algorithms, among the twelve, to use with respect to the type of entities or relations searched by the user.

Knowledge Base. The knowledge base contains the information used for the learning phase the truth finding procedure. These information include the true facts for some relations which have been learnt based on the feedbacks of the users. In addition, our knowledge base could be enriched with ground truth about some facts from reliable sources such as Wikipedia. Based on the knowledge base, our system has the ability to improve the accuracy of the truth finding process by learning about the best method to use or the best parameters, e.g., sources' accuracy scores, to consider for a better boostrapping of the process.

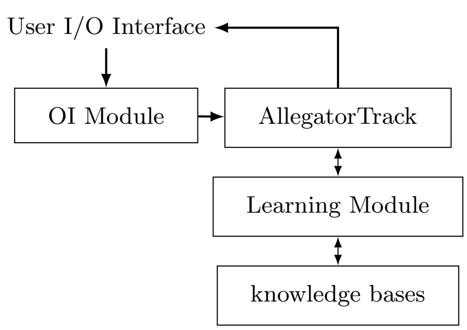

Figure 1: Architecture of our system

4.2 Demonstration Scenario

A given user that wants to interact with our system must do it through the search form. Through the search form, she (or he) provides her searched relation, e.g., "Where is born Barack Obama ?". The searched relation is then passed to the information extraction engine, TextRunner system in our case, which returns a set of answers considered to be relevant for the user's request. Each claim in the returned list is processed in order to extract the corresponding sources along a detailed description of the claim which we format in a certain manner. The set of sources and the formatted versions of all claims are then passed to the truth finding module which integrate all the claims and compute the most probable answer together with the reliability scores of participated sources for the searched relation. Finally, the output of the truth finding process is returned to the user. The user can also want to review the output of our system by definitively validiting it or not through its knwoledge of the modeled world. For example when the system has totally wrong, it may be interesting to get such a kind of feedbacks from the user in order to change the used method, as there are many available with our system, and to enhance the process for the further search about the same world. The user gives feedbacks using the option buttons on the left-hand side of the outputted claims or the text form. The feebacks given by the user is saved in knwoledge bases within our system for further processes.

5. CONCLUSION

[]

6. REFERENCES

[1] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open Information Extraction from the Web. *Commun. ACM*, 51(12) :68–74, December 2008.

[2] Naeemul Hassan, Chengkai Li, and Mark Tremayne. Detecting check-worthy factual claims in presidential debates. In *Proc. CIKM*, pages 1835–1838. ACM, 2015.

[3] Dalia Attia Waguih, Naman Goel, Hossam M. Hammady, and Laure Berti-Equille. AllegatorTrack : combining and reporting results of truth discovery from multi-source data. In *Proc. ICDE*, pages 1440–1443, 2015.

[4] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. TextRunner : Open Information Extraction on the Web. In *Proc. NAACL*, pages 25–26. Association for Computational Linguistics, 2007.