Active Truth Finding for relevant Information Retrieval

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

Open Web Information extraction systems like TextRunner [3] and popular Web search engines such as Google or Bind usually reply to users' search queries by returning a set of potential relevant answers. For some specific type of queries, the returned list might contain conflicting answers which make thing harder for the end-users to distinguish between the truth and the false. We demonstrate in the paper a system that processes answers outputted by open Web information extraction systems like TextRunner and provides the most probable answer using truth finding. Our system has also the capability to account for users' feedbacks, based on its knowledge of the correct instances for some searched relations, in order to improve the truth finding process.

1. INTRODUCTION

[**Lamine: Proposition of page allocation for the demo paper**]

- 1.25 pages –> abstract + introduction
- 1.5 pages –> Active Truth finding Process
- 1 pages –> Demonstration system
- 0.25 pages –> references

2. ACTIVE TRUTH FINDING PROCESS

2.1 Information Extraction Technique

2.2 AllegatorTrack Module

2.3 Active Truth Finding

3. OUR DEMONSTRATION SYSTEM

3.1 Architecture and GUI

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

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.

3.2 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

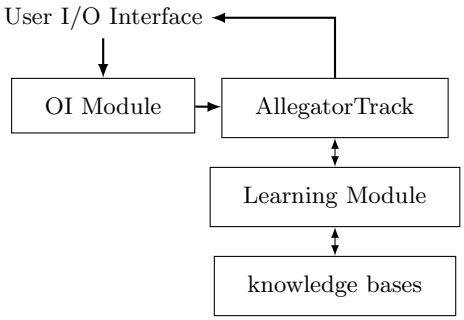

Figure 1: Architecture of our system

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.

4. CONCLUSION

□

5. 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] 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.

[3] 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, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.