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 are seeking to demonstrate in this paper the usefulness of truth discovery on large sets of *"factoid" claims* about real-world facts which are obtained when querying *open information extraction* (OpenIE) systems. These claims are usually extracted by information extractors from unreliable and conflicting Web sources. A "factoid" claim, e.g., Barack Obama was born in Kenya, often refers to a piece of information that is accepted as true, without any prior verification, because of its frequent redundancy over numerous sources. Multiple conflicting Web claims of such a type related to a specific real life fact are simultaneously available in practice. We draw, therefore, our attention on those conflicting claims returned by a common open Web information extraction system, *TextRunner* in our special case, as possible answers to users' queries. Concretely speaking, given a user input query (or key phrase), our main goal is to consider and improve the result of TextRunner[1] by running truth discovery in order to provide to the user the most reliable information. We next briefly present TextRunner system. Then, we detail how we format its output in such a that it fits the input of our truth discovering process.

Information extraction. We aim at analyzing conflicting claims about the same real-world facts from TextRunner. TextRunner [12, 2] is an OpenIE system relying on an unsupervised Web extraction process over a predefined set of Web corpus consisting of *Google*, *ClueWeb*, *News*, *Nell*, and *Wikipedia* corpus. Each particular corpus aggregates data from multiple other Web sources which are of various nature, for instance domain-specific Websites. Queries are sent to TextRunner by users through a form where they also can optionally specify their trusted corpus on the predefined set. When a user query arrives, TextRunner first finds the relevant sources in the set of corpus, and then it extracts each possible answer from them using natural language processing techniques and ontologies. To do so, it performs, for efficiency concerns, a single data-driven pass on the corpus to obtain the list of candidate relational tuples which might satisfy the arguments of the user input query. A typical user query in TextRunner is often about two real-world *entities* and a certain *relation* between them. As a consequence, such a user query q can be defined formally as a triplet (e_1, r, e_2) where e_1 and e_2 are real-world entities and r is a relation. The argument r specifies a possible relationship between the two given entities e_1 and e_2. In general, at least one among the three arguments is unkown, which captures a partial knowledge about the real world. In this context, TextRunner tries to find out the actual possible values of unkown arguments given known ones by querying the Web.

The outcome of TextRunner, given a user query, is indeed a set of candidate answers which is ranked according to the number of sources supporting each. TextRunner also enables to access, via Web hyperlinks, to the source and the document associated to each extracted answer. Our demonstration system (see Section 4) will follow these links and will extract the different sources of each potential answer

1. http://openie.allenai.org/

Un exemple d'extraction avec TextRunner
(a) TextRunner Extraction

Un exemple de triplés extraites
(b) Extracted Triplets

Figure 1: Data collection and formatting

for truth finding purposes.

Information processing. We consider a truth discovery process that takes, as input data, a set of claims in the form of quadruplets (claimID, source, claimQuery, claimValue) and infer, as outputs result, a *Boolean truth label* for each claim in which claimQuery is the query the claim is referred to and claimValue is the answer given by the source for the query. We detail below how these claims are inferred from the outcome of OpenIE TextRunner.

Assume a user query q about a real-word fact f_q and the set of n answers $v_1^q \ldots v_n^q$ returned by TextRunner for q. Let now denote by $\mathcal{S}_i^q$ the set of sources supporting each answer v_i^q, for $1 \leq i \leq n$. Recall that we extract this set of sources by following Web hyperlinks attached to answers by TextRunner and by using hand-written mapping rules. In addition note that for the same query, TextRunner returns only one answer per source, i.e., $\mathcal{S}_{c_i} \cap \mathcal{S}_{c_j} = \emptyset$ for all $i \neq j$ with $1 \leq i \leq j \leq n$. We have now, for the query q about the fact f_q, the set of claims together with the sources. In order to fit to the input of the truth finding process, we need to format the data collected from TextRunner about the facts in a certain manner. As a consequence, for the fact f_q, we therefore consider every extracted claim c_i and generate a triplet (f_q, c_i, s_j) for each source $s_j \in \mathcal{S}_{c_i}$. We finally obtain a collection of triplets $\{(f_q, c, s) \mid c_q \in \mathcal{C}, s \in \mathcal{S}_c\}$ as the final formatting of the output of TextRunner regarding a given user input query q about a fact f_q.

[**Lamine: Quelle types de requêtes souhaitons nous supporter?**]

[**Lamine: Si on considére chaque requête séparée, on se retrouve á traiter un seul claim á chaque fois. Généraliser á k requêtes utilisateurs?**]

[**Lamine: Peut être qu'il serait interessant d'avoir une idée de la distribution du nombre de conflits per query sur TextRunner**]

3. ACTIVE ENSEMBLING FOR TRUTH FINDING

We describe in this section our use of active ensembling in the context of truth finding for discovering gradually the optimal set of algorithms that together maximizes the accuracy of the process when users' feedbacks arrive into the system.

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

Our assumption. Ensembling should allow to perform truth discovery consistently well across datasets without having to determine a *priori* a suitable truth finding algorithm.

Ensembling, or commonly ensemble-based active learning, is a semi-supervised learning method that learns about an appropriate combinaison of multiple classifier types for a given task by interactively querying users (or other types of sources) for *labeled examples*.

In our setting, we are interesting on discovering the optimal combinaison among several possible truth finding algorithms which correspond to our classifier types. This combinaison is excepted to maximize the precision of the truth finding process on claims outputted by TextRunner as possible answers with respect to a user's query about a given real-world fact. A well known lack, e.g., as shown in [5, 9], of existing truth discovering algorithms is that they are mostly sensitive to particular application domains and data characteristics. As a consequence, there is no approach that outperforms the others on all types of datasets. Furthermore, the truth finding process is harder in practical scenarios in the sense that there usually exists no labeled examples, or *ground truth data*, against which one can evaluate the precision of the different algorithms in order to choose the more accurate one on data of interest. Fortunately, however, human being has often a certain knowledge background about certain real world facts. The user knowledge is a valuable source of labeled examples that should be harnessed as partial standard for evaluating the accuracy of our truth finding process. Indeed, even though a partial ground truth data could be obtained from the users, an optimal truth finding strategy change over time as we obtain more data from sources, for instance when claims are continuously extracted from Web sources by TextRunner. To tackle the aformentionned lacks of exsisting truth finding algorithms, especially in real applications like information extracted systems, we will use an active ensembling in truth finding in order to continuously discover an optimal hybrid truth finding approach when the extractor is gradually queried and users' feedbacks arrive. As we shall show later, we will actively involve users for labeled example in our ensemble-based learning process.

Determining the optimal sample of unlabeled items to send to the sources, e.g., users, for labels during an active learning problem is a challenging problem. Several query strategies, e.g., uncertainty sampling or query by committee, or Support vector machine (SVM) models have been proposed for the definition of such a optimal sample of data. Note that the type of the selected data along with the size of the sample are crucial for the effectiveness of the learning procedure; we defer to [8] for more details about active machine learning. In this study, we have used query by committee strategy in which an ensemble of hypotheses is learned and examples that cause maximum disagreement amongst this committee (with respect to the predicted truth) are selected as the most informative. points for which the "committee" disagree. Query by committee is known to be a very effective active learning approach that has been successfully applied to different classification problems.

[**Lamine:** **J'ai juste mentionné "query by committee" in guise d'exemple. La stratégie utilisée doit être choisie.**] [**Lamine:** **Basic QBC or Bagging or Boosting ?**]

We use and compare twelve well established truth finding algorithms in the literature, which we cluster in different classes according to their specificities. We briefly present each class of considered truth discovering algorithms in the following.

1. **Iterative techniques:** TruthFinder [13], Cosine, 2-Estimates and 3-Estimates [3], AccuNoDep [1]
2. **EM based techniques:** MLE [11], LTM [14], SimpleLCA and GuessLCA [7]
3. **Dependency detection based techniques:** Depen, Accu, and AccuSim [1]

[**Lamine:** **La classification des algorithmes est juste une proposition. Peut être qu'il existe une meilleure classification.**]

3.2 Active Learning Process

- notre approche que l'on défend ici dans la démo est semi supervisée en impliquant de l'utilisateur 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

We rely on an ensemble-based semi-supervised learning process for discovering an hybrid, i.e., an optimal ensemble of truth finding algorithms for information extraction systems. Let denote by $\mathcal{Q}$ the set of successive user queries processed by the information extraction system. For each query q in $\mathcal{Q}$ about the fact f_q we have the corresponding sets of claims $\mathcal{C}_q$ returned by the extractor. We refer to the entire set of all claims by $\mathcal{C}$ regarding $\mathcal{Q}$. We assume that $\mathcal{C}$ contains labeled and unlabeled claims where labeled claims, corresponding to our partial ground truth, are those for which we know whether they are correct or not by querying the user. In contrast, we do not know yet the truth about unlabeled claims and would like to discover by using the best ensemble of truth finding algorithms. We refer respectively to labeled and unlabeled set of claims by $\mathcal{C}^{\mathsf{L}}$ and $\mathcal{C}^{\mathsf{U}}$. Given a base learning algorithm X, a number k of fixed act iterations, and a fixed size m of a sampling, we perform truhth finding with ensembling on our set of truth finding algorithms as follows.

1. We first train our ensemble of truth finding algorithms (representing here our set of classifiers) on the current set of labeled claims $\mathcal{C}^{\mathsf{L}}$ [**Lamine:** **How the base learning algorithm should be implemented?**]
2. We then pass our set of unlabeled claims in $\mathcal{C}^{\mathsf{U}}$ to the ensemble of truth finding algorithms
3. Each algorithm in the current ensemble predicts the label of each claim
4. The claims that induce the most label prediction disagreement are queried for their labels and added to the training set
5. We select a subset T of the m claims that induce the most label prediction disagreement
6. We request the labels of these m claims to the user
7. We lastly remove claims in T from $\mathcal{C}$ and add them together with their acquired truth labels to $\mathcal{C}^{\mathsf{L}}$.

We repeat the process above k times or until the ensemble meets some pre-defined criteria, e.g. when the size of the ensemble reaches a certain pre-defined value. At the end of the active learning process, the prediction, i.e., truth discovery, is made by taking the majority vote of the resulting ensemble members.

4. OUR DEMONSTRATION SYSTEM

We describe in this section our system for combining truth discovring and information extraction with active ensembling. We first present the architecture of our system by giving its different modules. Then, we provide a typical demonstration scenario of a user interacting with our system.

4.1 GUI and System architecture

The architecture of our demonstration system, given in Figure 2, comprises the following three main components.

Graphical user 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 Module. It corresponds to AllegatorTrack which contains twelve truth finding algorithms with different accuracy according to the types of claims and the characteristics of sources.

Active Ensembling 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.

Repository of Labeled Facts. 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.

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,

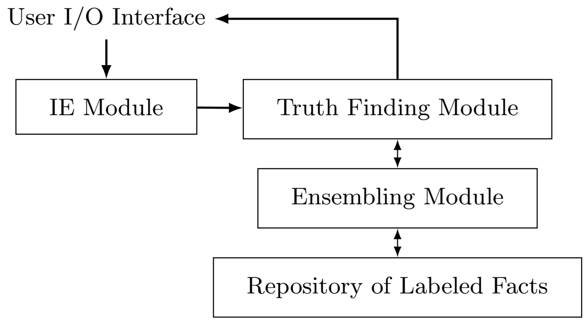

Figure 2: Architecture of our system

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

[**Lamine: L'utilisateur peut faire une erreur sur l'étiquette de certains claims. Comment capturer ce phénoméne?**] []

6. REFERENCES

[1] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. Integrating conflicting data: The role of source dependence. *PVLDB*, 2(1):550–561, 2009.

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

[3] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. Corroborating Information from Disagreeing Views. In *WSDM*, pages 131–140, 2010.

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

[5] Xian Li, Xin Luna Dong, Kenneth Lyons, Weiyi Meng, and Divesh Srivastava. Truth finding on the deep web: Is the problem solved? *Proc. VLDB Endow.*, 6:97–108, December 2012.

[6] Zhenyu Lu, Xindong Wu, and J.C. Bongard. Active learning through adaptive heterogeneous ensembling. *IEEE Transactions on Knowledge and Data Engineering*, 27:368–381, Feb 2015.

[7] Jeff Pasternack and Dan Roth. Latent Credibility Analysis. In *WWW*, pages 1009–1020, 2013.

[8] Burr Settles. *Active Learning*. Number 114. Morgan & Claypool Publishers,, June 2012.

[9] Dalia Attia Waguih and Laure Berti-Equille. Truth discovery algorithms: An experimental evaluation. *CoRR*, abs/1409.6428, 2014.

[10] 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.

[11] Dong Wang, Lance M. Kaplan, Hieu Khac Le, and Tarek F. Abdelzaher. On Truth Discovery in Social Sensing: a Maximum Likelihood Estimation Approach. In *IPSN*, pages 233–244, 2012.

[12] 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.

[13] Xiaoxin Yin, Jiawei Han, and Philip S. Yu. Truth Discovery with Multiple Conflicting Information Providers on the Web. *TKDE*, 20(6):796–808, 2008.

[14] Bo Zhao, Benjamin I. P. Rubinstein, Jim Gemmell, and Jiawei Han. A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration. *PVLDB*, 5(6):550–561, 2012.