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 $v_1^q \dots v_n^q$ 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. $\mathcal{S}_i^q \cap \mathcal{S}_j^q = \emptyset$ for all $i \neq j$ with $1 \leq i \leq j \leq n$. We have the set of potential answers along with their respective sets of sources for the user query q about the fact f_q. We can now proceed to the generation of the corresponding claims. To do so, we consider each answer v_i^q, for each $1 \leq i \leq n$, and loop on its set of sources $\mathcal{S}_i^q$. For each source $s \in \mathcal{S}_i^q$, we create a new claim $(\mathsf{claimID}, s, q, v_i^q)$ with an automatically generated unique claim identifier claimID, a claim source source corresponding to s, q as the claim query claimQuery, and a claim value claimValue equals to v_i^q. In other words, we create as many claims as the number of sources of a given answer from TextRunner. The total number of generated claims for the query q being equals to $\sum_{1 \leq i \leq n} |\mathcal{S}_i^q|$.

Let us assume the set of m successive user queries $q_1, \dots, q_m$. We finally suppose that when the query is successively issued in TextRunner and answered by the system, the aforementioned formatting procedure transforms its set of possible answers to the corresponding set of claims. We refer to the overall set of resulting claims for queries $q_1, \dots, q_m$ with $\mathcal{C}$.

[**Lamine: Types de requêtes á supporter?**]

[**Lamine: Traitement de requêtes en batch ou traitement une par une?**]

[**Lamine: Regarder la distribution du nombre de conflits per query sur TextRunner pour mieux motiver l'utilité du truth finding?**]

3. ENSEMBLING FOR TRUTH FINDING

We present in this section an adaptive truth finding approach which uses active ensembling in order to adaptively learn about an optimal set of truth finding algorithms that outperforms any individual technique on any given dataset. Our learning approach will actively involve users for the correct labels (or answers) of a sample of queries that cause maximal disagreement amongst our classifiers.

3.1 Ensemble-Based Active Learning

- 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-based active learning, or commonly ensembling, is a semi-supervised learning approach that tries to figure out an optimal ensemble of classifiers for a given classification problem by actively querying an oracle, e.g., a human being, about the labels of a sample of data items. Ensembling, thereby, enables to perform classification consistently well across datasets without having to determine a *priori* a suitable classifier type.

In our context, the truth finding algorithms correspond to our set of classifiers. The underlying *binary classification* problem consists of assigning the correct truth label to a set of claims about given user queries. Indeed, a truth finding algorithm is formally a mapping $\mathsf{TF} : \mathcal{C} \mapsto \{\mathsf{true}, \mathsf{false}\}$ which associates to each claim in $\mathcal{C}$ either true or false. A good truth finding algorithm provides predictions that are close to the actual world. Unfortunately, a well known property, e.g., as shown in [5, 9], of existing truth discovering algorithms remains their sentivity to certain application domains or datasets. As a consequence, there is no actual approach that outperforms the others on all types of datasets. On the other hand, truth finding is hard in practical scenarios because there is often no prior knowledge guiding to the selection, beforehand, of an optimal algorithm, in particular when the context is dynamic. More importantly, a large set of labeled examples (or ground truth) for evaluating the precisions of the algorithms is expensive to obtain in real applications.

In general, human being has a certain background knowledge about some real-world facts. Such a knowledge can serve as a valuable and inexpensive source of labels for a rather reasonable number of data items. However, having this partial ground truth from users is not sufficient in order to definitively decide about an optimal truth finding strategy because it can change over time as we obtain more information from sources, e.g. when claims are continuously extracted by TextRunner for answering new incoming queries. Therefore, there is a need for an adaptive approach able to dynamically figure out the optimal truth finding strategy when users' feedbacks and new knowledge about the world are available. We believe that active ensembling should be helpful to this end.

We put forward and demonstrate an approach which combines truth discovery and open information extraction with ensemble-based active learning for adaptively learning about the optimal ensemble of truth finding algorithms when the OpenIE system is gradually querying and labeled examples from users are available. As we shall show later, we will actively involve users to obtain the truth about a sample of particular facts during the learning process. The way this sampling is defined and chosen is crucial for the effectiveness

of the active learning. Several sample selection strategies, e.g., random sampling, query by committee, or support vector machine models, have been proposed for the definition of the type of selected data items along the size of the sample; we defer to [8] for more details about active machine learning. In this study, we use *query by committee* (QBC) for ensemble-based active learning. QBC states that the best data items to select for labels are those that cause the *maximal disagreement* among the predictions of an ensemble of diverse but partially accurate classifiers during active learning. Furthermore, we seek to provide an adaptive active learning by looking for an optimal ensemble given a larger set of input classifiers. [**Lamine: Peut être qu'il y a mieux que QBC ?**]

To learn about an optimal ensemble from a diverse set of classifiers, we have considered twelve well established truth finding algorithms in the literature, having three different types according to their specificities. Note that diversity offers better result in active learning than using homogenoeus classifiers (see [6]). We briefly present each considered class of 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: Peut être qu'il existe une meilleure classification ?**]

3.2 Truth Finding with Active Ensembling

– 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 present an adaptive truth finding algorithm based on active ensembling in order to learn about an optimal ensemble over a set of existing truth finding algorithms, on which one can efficiently find the truth for the output of OpenIE systems. Our approach first obtains from the learning procedure intuitions about the best algorithm to use for each incoming fact and then it performs the union of the result of the ensemble of best returned truth finding algorithms for a set of facts. The best truth finding algorithm for a fact, i.e., the algorithm that has the highest chance to reliability discover the truth among a set of candidates claims about this fact, will be found by the learner by evaluating the accuracy of each competing technique on labeled claims from the user. We sketch in the following the procedure by assuming that all the input truth finding algorithms are used with their optimal initial parameters which are deemed known beforehand.

Our ensemble-based learning process relies on QBC for label querying and aims at finding an optimal combinaison of the result of the different truth finding algorithm involved in the process. Such a learning process is iterative. Consider the set of all claims $\mathcal{C}$ and the set of labeled claims $\mathcal{C}_{GT}$. The claims in $\mathcal{C}$ are progressively obtained and processed, like in a streaming manner, as from the information extraction system as long as it receives queries. The set $\mathcal{C}_{GT}$, initially empty, contains claims whose labels are known for sure by asking the user. We also assume a stop criteria (e.g., a predefined number of iterations or accuracy changes between two iterations) for our iterative learning algorithm for truth finding and the set of optimal initial parameters for each truth finding algorithm. The algorithm starts by evaluating the truth finding algorithms on the available unlabeled claims in $\mathcal{C}$ for determining the prediction of each technique. It then compare the set of the different algorithms on each set of claims about the same query (or fact) and determines the queries causing the maximal disagreement among the members of the committee. Those queries are determined by computing the vote entropy of each query. The algorithm requests to the user labels for the set of candidates claims of those queries. Once it obtained labels from the user, it adds them into $\mathcal{C}_{GT}$ and then discards them from $\mathcal{C}$.

At each iteration, the algorithm chooses exactly one set of claims about the same query for asking their labels to the user. Once these labels are acquired from the user, the algorithm adds the claims along with their truth labels into the training set, adapts the ensemble of truth finding algorithms for an optimal one, discards the old one and trains the new ensemble against the actual training set. We detail the learning procedure below. [**Lamine: Une esquisse du process d'apprentissage active**]

As a consequence and similarly to [6], we rely on an iterative active ensembling process which considers the set of claims a training set of labeled claims $\mathcal{C}_{TR}$, testing set of claims , an ensemble of truth finding algorithms, an adaptation set which is used for adapting the configuration of the ensemble, a and a stopping criteria .

1. The algorithm starts with an *initialization phase* initial initial values of parameters are set, an initial training set and ensemble of truth finding algorithm are defined. Such an ensemble is trained on this initial training set.
2. The algorithm pursues by giving the claims in $\mathcal{C}$ to the truth finding algorithm in the current ensemble for label predictions and then it records the vote entropy of each query, thereby the underlying fact, according to the predictions of the committee.
3. It chooses the set of claims associated to the query having the maximum vote entropy for label querying
4. The set of acquired truth labels, together with the corresponding claims, are added into the training set and then discarded from $\mathcal{C}$.
5. At this stage the algorithm uses an *adaptation procedure* in order to tune the ensemble of truth finding algorithm towards an optimal one.
6. The newly generated ensemble is trained on the updated training set of labeled claims.

The steps 2–6 of the active learning algorithm are repeated until the stop criteria is satisfied. At the end of the active learning process, the truth discovery is finally realized by performing a majority vote on the predictions of the final inferred ensemble.

[**Lamine: Esquisse de la méthode d'adaptation (á finir)**] In the adaptation step, the algorithm tries to determine the optimal ensemble of truth finding algorithms which will guarantee the highest prediction accuracy at the end. To do so, we assume that our algorithm starts with an initial ensemble including one algorithm from each class.

At each iteration, the adaptation process creates variants of the current ensemble by discarding one algorithm from the ensemble.

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, 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

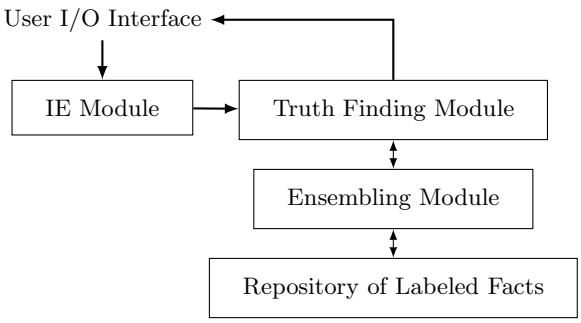

Figure 2: Architecture of our system

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.