User Guidance for Efficient Fact Checking

Nguyen Thanh Tam [1], Matthias Weidlich [2], Hongzhi Yin [3], Bolong Zheng [4],
Nguyen Quoc Viet Hung [5], Bela Stantic [5]

[1] École Polytechnique Fédérale de Lausanne, Switzerland, [2] Humboldt-Universität zu Berlin, Germany,
[3] University of Queensland, Australia, [4] Aalborg University, Denmark [5] Griffith University, Australia

ABSTRACT

The Web constitutes a valuable source of information. In recent years, it fostered the construction of large-scale knowledge bases, such as Freebase, YAGO, and DBpedia. The open nature of the Web, with content potentially being generated by everyone, however, leads to inaccuracies and misinformation. Construction and maintenance of a knowledge base thus has to rely on fact checking, an assessment of the credibility of facts. Due to an inherent lack of ground truth information, such fact checking cannot be done in a purely automated manner, but requires human involvement.

In this paper, we propose a comprehensive framework to guide users in the validation of facts, striving for a minimisation of the invested effort. Our framework is grounded in a novel probabilistic model that combines user input with automated credibility inference. Based thereon, we show how to guide users in fact checking by identifying the facts for which validation is most beneficial. Moreover, our framework includes techniques to reduce the manual effort invested in fact checking by determining when to stop the validation and by supporting efficient batching strategies. We further show how to handle fact checking in a streaming setting. Our experiments with three real-world datasets demonstrate the efficiency and effectiveness of our framework: A knowledge base of high quality, with a precision of above 90%, is constructed with only a half of the validation effort required by baseline techniques.

PVLDB Reference Format:
Nguyen Thanh Tam, Matthias Weidlich, Hongzhi Yin, Bolong Zheng, Nguyen Quoc Viet Hung, Bela Stantic. Efficient User Guidance in Fact Checking. *PVLDB*, 12(xxx): xxxx-yyyy, 2019.
DOI: https://doi.org/TBD

1. INTRODUCTION

Extracting factual knowledge from Web data plays an important role in various applications. For example, knowledge bases such as Freebase [3], YAGO [7] and DBpedia [1] rely on Wikipedia to extract entities and their relations. These knowledge bases store millions of facts, about society in general as well as specific domains such as politics and medicine. Independent of the adopted format to store facts, extraction of factual knowledge first yields candidate facts (aka claims), for which the credibility needs to be assessed. Given the open nature of the Web, where content is potentially generated by everyone, extraction of claims faces inaccuracies and misinformation. Hence, building a knowledge base from Web sources does not only require conflict resolution and data cleansing [23], but calls for methods to ensure the credibility of the extracted claims, especially in sensitive domains, such as healthcare [48].

To assess the credibility of claims, automated methods rely on classification [41] or sensitivity analysis [66]. While these methods scale to the volume of Web data, they are hampered by the inherent ambiguity of natural language, deliberate deception, and domain-specific semantics. Consider the claims of 'the world population being 7.5 billion' or 'antibiotics killing bacteria'. Both represent common-sense facts. Yet, these facts have been derived from complex statistical and survey methods and, therefore, cannot easily be inferred from other basic facts.

When relying on accurate facts, incorporating manual feedback is the only way to overcome the limitations of automated fact checking. However, eliciting user input is challenging. User input is expensive (in terms of time and cost), so that a validation of all claims is infeasible, even if one relies on a large number of users (e.g., by crowdsourcing) and ignores the overhead to resolve disagreement among them. Also, claims are not independent, but connected in a network of Web sources. An assessment of their credibility thus requires effective propagation of user input between correlated claims. Finally, there is a trade-off between the precision of a knowledge base (the ratio of credible facts) and the amount of user input: The more claims are checked manually, the higher the precision. However, user input is commonly limited by some budget.

This paper presents a comprehensive framework for guiding users in fact checking, adopting a pay-as-you-go approach. We present a novel probabilistic model that enables us to reason on the credibility of facts, while new user input is continuously incorporated. By (i) inferring the credibility of non-validated facts from those that have been validated, and by (ii) guiding a user in the validation process, we reduce the amount of manual effort needed to achieve a specific level of result precision. Credibility inference and user guidance are interrelated. Inference exploits mutual reinforcing relations between Web sources and claims, which are further justified based on user input. Moreover, a user is guided based on the potential effect of the validation of a claim for credibility inference.

Efficient user guidance further requires to decide: (i) when to terminate validation to avoid wasting resources on marginal improvements of the quality of the knowledge base; (ii) how to group claims for batch processing to reduce the impact of set-up costs in validation (a user familiarising with a particular domain); and (iii) how to handle continuous arrival of new data to avoid redundant computation. Our novel model enables us to address these aspects.

Proceedings of the VLDB Endowment, Vol. 12, No. xxx
Copyright 2018 VLDB Endowment 2150-8097/18/10... $ 10.00.
DOI: https://doi.org/TBD

Our contributions are summarised as follows:

- *Approach to Guided Fact Checking:* §2 formalises the setting of fact checking and, based thereon, formulates the problem of effort minimisation. We further introduce an iterative approach to guide a user in the validation process and highlight requirements for its instantiation.
- *Probabilistic credibility inference:* §3 addresses the need for a method to reason on the credibility of facts. We introduce a probabilistic model for fact checking, based on Conditional Random Fields, and show how to perform incremental inference based on user input. Aiming at pay-as-you-go validation, we show how to derive a trusted set of facts based on our model.
- *Probabilistic user guidance:* §4 presents strategies to guide users, i.e., to select the claims for which validation is most beneficial. These strategies target the reduction of uncertainty in our probabilistic model for fact checking.
- *Complete validation process:* §5 combines our mechanisms for credibility inference and user guidance to obtain a comprehensive validation process. We also show how to achieve robustness against erroneous user input.
- *Methods for effort reduction:* §6 introduces techniques for early termination of the validation process and batch selection. The former is based on signals that indicate convergence of our probabilistic model and, thus, of the quality of the derived knowledge base. The latter selects groups of claims for validation based on the benefit of their joint validation. Since this selection problem turns out to be intractable in practice, we propose a greedy top-k algorithm, which comes with performance guarantees.
- *Streaming fact checking:* §7 shows how to handle continuously arriving data by an adaptation of our validation process that features stochastic approximation and reuse of model parameters.

We evaluate our techniques with three large-scale datasets (§8) of real-world claims. We demonstrate low response times for claim selection (<0.5s) and high effectiveness of guiding users in their validation efforts. To obtain a knowledge base of high quality (>90% precision), only a half of the effort of baseline techniques is required. Finally, we review related work (§9) and conclude (§10).

2. GUIDED FACT CHECKING

2.1 Setting

We model the setting of fact checking by means of a set of data sources $\mathcal{S} = \{s_1, \ldots, s_u\}$, a set of documents $\mathcal{D} = \{d_1, \ldots, d_m\}$, and a set of candidate facts, or short claims, $\mathcal{C} = \{c_1, \ldots, c_n\}$. A source could be a user, a website, a news provider, or a business entity. It provides multiple documents, each often being textual (e.g., a tweet, a news item, or a forum posting) and involving a few claims. The representation of a claim (e.g., unstructured text or an RDF triple) is orthogonal to our model. However, a claim can be referenced in multiple documents, it depends on a specific process for information extraction how the link between claims and documents is established (see §8.1).

A claim $c \in \mathcal{C}$ represents a binary random variable, where $c = 1$ and $c = 0$ denote that the claim is credible or non-credible, respectively. In fact checking, however, these values are not known, so that we consider a probabilistic model P, where $P\,(c = 1)$, or $P\,(c)$ for short, denotes the probability that claim c is credible. Combining the above notions, the setting of fact checking is a tuple $Q = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P \rangle$, also referred to as a *probabilistic fact database*.

A knowledge base is constructed from such a database by deriving a trusted set of facts. We formalise this construction by a grounding function $g : \mathcal{C} \rightarrow \{0, 1\}$, labelling claims as credible ($g(c) = 1$) or non-credible ($g(c) = 0$).

In fact checking, claims are validated manually by a user, which is represented by a binary model of user input. A claim c is either confirmed as credible, which yields $P(c) = 1$, or labelled as non-credible, so that $P(c) = 0$.

As an example, consider the *Snopes* dataset [9], a collection of 4856 claims derived from 80421 documents of 23260 sources, such as news websites, social media, e-mails, etc. For instance, this dataset comprises the claim that *eating turkey makes people especially drowsy*. This claim can be found in documents of various Web sources, among them earthsky.org [2], webmd.com [6], and kidshealth.org [4]. In the Snopes dataset, claims have been validated by expert editors, which corresponds to the user input in our model. It labels the aforementioned example claim as non-credible [5].

2.2 Effort Minimisation

Adopting the above model, the grounding g to derive a trusted set of facts is partially derived from user input. However, manual validation of claims is expensive, in terms of user hiring cost and time. User input is commonly limited by an effort budget, which leads to a trade-off between validation accuracy and invested effort.

Going beyond this trade-off, we aim at minimising the user effort invested to reach a given validation goal. We consider fact checking as an iterative process with a user validating the credibility of a single claim in each iteration. This process halts either when reaching a validation goal or upon consumption of the available effort budget. The former relates to the desired result quality, e.g., a threshold on the estimated credibility of the grounding. The latter defines an upper bound for the number of validations by a user and, thus, iterations of the validation process.

Formally, given a probabilistic fact database $\langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P \rangle$, fact checking induces a *validation sequence*, a sequence of groundings $\langle g_0, g_2, \ldots, g_n \rangle$ obtained after incorporating user input as part of n iterations of a validation process (i.e., any g_i is a prediction of the model). Given an effort budget b and a validation goal Δ, a sequence $\langle g_0, g_1, \ldots, g_n \rangle$ is *valid*, if $n \leq b$ and g_n satisfies Δ. Let $\mathcal{R}(\Delta, b)$ denote a finite set of valid validation sequences that can be created by instantiations of the validation process. Then, a validation sequence $\langle g_0, g_1, \ldots, g_n \rangle \in \mathcal{R}(\Delta, b)$ as *minimal*, if $n \leq m$ for any validation sequence $\langle g'_0, g'_1, \ldots, g'_m \rangle \in \mathcal{R}(\Delta, b)$.

Problem 1 (Effort Minimisation)**.** *Let* $\langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P \rangle$ *be a probabilistic fact database and* $\mathcal{R}(\Delta, b)$ *a set of valid validation sequences for an effort budget* b *and a goal* Δ*. The problem of* effort minimisation in fact checking *is the identification of a minimal sequence* $\langle g_0, g_1, \ldots, g_n \rangle \in \mathcal{R}(\Delta, b)$*.*

The validation goal could be the precision of the final grounding g_n, estimated by cross validation. Note that, in theory, Problem 1 could have no solution—the effort budget may be too small or the validation goal may be unreachable. However, for practical reasons, there needs to be a guarantee that the validation process terminates.

Solving Problem 1 is challenging, mainly for two reasons. First, claims are not independent, but subject to mutual reinforcing relations with Web sources and documents. Consequently, the validation of one claim may affect the probabilistic credibility assessment of other facts. Second, the problem is computationally hard: Finding an optimal solution quickly becomes intractable, since all permutations of all subsets (of size $\leq b$) of claims would have to be explored.

2.3 Outline of the Validation Process

To address the problem of effort minimisation, we argue that a user shall be guided in the validation of claims. In essence, user input shall be sought solely on the 'most promising' unverified facts, i.e., those for which manual validation is expected to have the largest impact on the estimated credibility of the resulting grounding.

Let $\langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P \rangle$ be a probabilistic fact database. Our validation process continuously updates the grounding g to validate claims in a pay-as-you-go manner, by:

(1) *selecting* a claim c for which feedback shall be sought;
(2) *eliciting* user input on the credibility of c, which either confirms it as credible or labels it as non-credible;
(3) *inferring* the implications of user input on the probabilistic credibility model P;
(4) *deciding* on the grounding g that captures the facts that are assumed to be credible.

In the above process, steps (1), (3), and (4) need to be instantiated with specific methods. An example for a straight-forward instantiation would be a validation process that:

- *selects* a claim c randomly for validation;
- limits the *inference* to claim c, setting either $P(c) = 1$ or $P(c) = 0$, not changing $P(c')$ for any claim $c' \neq c$;
- *decides* that a claim c is credible, $g(c) = 1$, if and only if it holds $P(c) \geq 0.5$.

In the remainder, we present methods for a more elaborated instantiation of the above process. We introduce a probabilistic model for fact checking that captures the mutual reinforcing relations between Web sources and claims. This enables us to *infer* the implications of user input beyond the claims that have been validated, and based thereon, *decide* on the grounding while incorporating the relations between sources and claims. Also, the model enables conclusions on the claims that shall be *selected*. Unverified claims for which validation is most beneficial for the inference will be chosen. Our model further helps to identify suspicious user input, i.e., claims that may have been validated by mistake.

We then address aspects of practical relevance, which are not captured in Problem 1. Validation may converge *before* the validation goal is reached and the effort budget has been spent. If so, further user input leads to diminishing improvements of the quality of the grounding and the validation process may be terminated. We show how our model enables the detection of such scenarios by decision-support heuristics.

In practice, users that validate claims face significant set-up costs, implied by the need to familiarise with claims of a particular domain. It therefore increases user convenience and efficiency if the validation process considers a batch of claims per iteration. We support such batching by a greedy top-k strategy to select a set of claims with a high *joint* benefit for credibility inference.

Moreover, in many applications, new sources, documents, and claims arrive continuously. We thus illustrate how the above process can be lifted to a streaming setting by exploiting online algorithms for inference and reusing parameters of our underlying model.

3. CREDIBILITY INFERENCE

This section presents a probabilistic model for fact checking (§3.1), before turning to mechanisms for incremental inference (§3.2) and the instantiation of a grounding (§3.3).

3.1 A Probabilistic Model for Fact Checking

Sources of uncertainty. Claims are assessed by means of documents from Web sources. These documents are encoded using a set of features. We abstract from the specific nature of these features, but take into account that the trustworthiness of a source and the language quality of a document have a strong influence on the credibility of the claims. We capture these features as follows. A source $s \in \mathcal{S}$ is associated with a feature vector $\langle f_1^{\mathcal{S}}(s), \ldots, f_{m_{\mathcal{S}}}^{\mathcal{S}}(s) \rangle$ of $m_{\mathcal{S}}$ source features. In the same vein, $\langle f_1^{\mathcal{D}}(d), \ldots, f_{m_{\mathcal{D}}}^{\mathcal{D}}(d) \rangle$ is a vector of $m_{\mathcal{D}}$ document features, assigned to each document $d \in \mathcal{D}$.

Features of sources and documents interact with each other, and with the credibility of claims. A claim's credibility depends on both, the trustworthiness of the source and the language quality of the document, which we call a *direct relation*. A claim is more likely to be credible, if it is posted by a trustworthy source using objective language. Yet, the intentions of a source, and thus its trustworthiness, may change over different contexts and hence documents. Therefore, we also reason about the credibility of claims via an *indirect relation*, exploiting that documents of different sources may refer to the same claim. For example, a source disagreeing with a considered credible by several sources shall be regarded as not trustworthy.

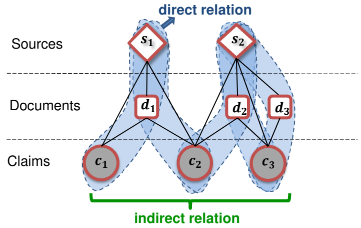

Figure 1: Relations in a probabilistic fact database.

The Conditional Random Field model. To model these relations, and eventually derive the assignment of credibility probabilities, we rely on a Conditional Random Field (CRF) [25], see Fig. 1. We construct a CRF as an undirected graph of three sets of random variables, $S, D, \mathcal{C}$ for sources, documents, and claims. Here, S and D are sets of real-valued variables that represent trustworthiness of sources and language quality of documents, respectively, based on the aforementioned features. Set $\mathcal{C}$ is the set of binary variables introduced in §2.1, each variable representing a claim's credibility. Direct relations are captured by relation factors in the CRF, also called cliques since they always involve three random variables (source, document, claim). Any random variable can be part in multiple cliques, reflecting the indirect relations. This implies a factorization of cliques to compute the joint probability distribution.

In this model, S and D are observed variables. As an output variable, we consider a categorical variable C that represents credibility configurations of claims. A possible value o of C, called configuration, is an assignment $o : \mathcal{C} \rightarrow \{0, 1\}$, such that each variable $c \in \mathcal{C}$ is assigned the value $o(c)$. Considering these variables, the model likelihood is expressed in the form of a conditional distribution, tailored from the generic form of a CRF [25]:

$$Pr(C = o \mid D, S; W) = \frac{1}{Z} \prod_{\pi = \{c,d,s\} \in \Pi} \phi(c = o(c), d, s; W_\pi) \quad (1)$$

where Π is the set of all cliques in the CRF; c, d, s are the claim, document, and source of a clique π, respectively; $Z = \sum_{c \in \mathcal{C}} \prod_{\pi \in \Pi} \phi(c = o(c), d, s; W_\pi)$ is a normalisation constant to ensure that the probabilities over all configurations of C sum up to one; and $W = \bigcup_{\pi \in \Pi} W_\pi$ is the set of model parameters controlling the effects of individual features. Using this model, we shall compute the conditional distribution of C, given the source and document features. This is realised by the log-linear model (aka logistic regression) that expresses the log of a potential function as a linear combination of features, instantiated from its generic form [25]:

$$\log\ \phi(c = o(c), d, s; W_\pi) = w_{\pi, o(c)} + \sum_{t=1}^{m_{\mathcal{D}}} w_{\pi,t}^{\mathcal{D}} \times f_t^{\mathcal{D}}(d) + \sum_{t=1}^{m_{\mathcal{S}}} w_{\pi,t}^{\mathcal{S}} \times f_t^{\mathcal{S}}(s). \quad (2)$$

Hence, we have different weights for each configuration of C and $W_\pi = \{w_{\pi,0}, w_{\pi,1}, w_{\pi,t}^{\mathcal{D}}, w_{\pi,t}^{\mathcal{S}}\}$ is the set of all weights.

The above formulation is motivated by the CRF being a special case of log-linear models, which, extending logistic regression, are suitable for structured learning tasks [38, 25]. In our setting, the data has an internal structure via the relations between sources, documents, and claims. Exploiting these relations, however, means that the inference of model parameters becomes complex. Hence, the potential function needs to be computationally efficient to enable user interactions in the validation process. A log-linear model enables efficient computation, while, at the same time, provides a comprehensive model, in which the features of sources and documents are discriminative indicators for the credibility of the related claims. The weights enable tuning of feature importance, as features vary between applications and shall be learned from labelled data.

Handling opposing stances. Documents may link the same claim with opposite stances—support or refute it [28]—and a source is considered trustworthy, if it refutes an incorrect claim. A model that only captures that a claim is part of a document would neglect this aspect. Yet, incorporating such information via a new type of random variable would mean that the number of variables is larger than or equal to the number of documents, which is much larger than the number of claims (see §8). We therefore introduce an opposing variable $\neg c$ for each claim c. Then, model complexity increases only slightly: Configurations of C include opposing claims, W contains a doubled number of parameters, and any document connects only to the positive or negative variable of a claim. As c and $\neg c$ cannot have the same credibility value, we enforce a non-equality constraint:

$$Pr(c, \neg c') = \begin{cases} 0 & \text{if } c = c' \\ Pr(c, \neg c' | D, S; W) & \text{otherwise.} \end{cases} \tag{3}$$

3.2 Incremental Inference with User Input

Using the above formalisation, we further distinguish the set $\mathcal{C}^L \subseteq \mathcal{C}$ of validated, or labelled, claims. It contains all claims c for which, based on user input, we set $P(c) = 1$ in the probabilistic fact database. In the same vein, $\mathcal{C}^U = \mathcal{C} \setminus \mathcal{C}^L$ is the set of unlabelled claims. Based thereon, we define restricted variants of the categorical random variable C that represents credibility configurations of claims: C^U and C^L are variables for configurations involving solely the unlabelled claims of $\mathcal{C}^U$ or the labelled claims of $\mathcal{C}^L$, respectively. Then, we need to solve the following optimisation problem to infer model parameters (as usual, $Pr(X)$ is the probability of one value of a categorical random variable X), derived from the principle of maximum likelihood [25]:

$$\begin{aligned} W^* &= \arg\max_W \ \log \ Pr(C^L \mid D, S; W) && (4) \\ &= \arg\max_W \ \log \sum_{C^U} Pr(C^L, C^U \mid D, S; W). && (5) \end{aligned}$$

The log-likelihood optimisation is convex, since the logarithm is monotonically increasing and the probability distribution is in exponential form. However, the problem becomes intractable due to the exponential number of configurations to consider for the random variable C^U. Moreover, upon receiving new user input, $\mathcal{C}^L$ and $\mathcal{C}^U$, and hence C^L and C^U change, so that re-computation is needed.

Requirements for model inference. To be useful in our setting, an inference algorithm must meet two requirements. First, user input on correspondences should be a first class citizen. By propagating which claims have been validated, credibility probabilities can be computed for claims for no input has been sought so far. Second, each iteration of the validation process changes the credibility of claims only marginally. Hence, inference should proceed incrementally and avoid expensive re-computation of the credibility probabilities and model parameters in each iteration.

Existing inference algorithms. Various inference algorithms have been proposed in the literature. Yet, none of them meets the aforementioned requirements. Traditional CRF models, such as [54], operate in a static manner, in which model parameters are inferred from a fixed set of labelled data by methods that incur high computational effort (e.g., gradient descent or trusted region methods). Hence, credibility probabilities and model parameters in our model would be computed from scratch every time new user input arrives. Moreover, the instantiation of a grounding based on this model requires another pass over the whole data. This makes it not suitable for interactive validation process considered in our work.

***iCRF* algorithm.** In the light of the above, we propose a novel incremental inference algorithm, $iCRF$, which adopts the view maintenance principle by maintaining a set of Gibbs samples over time. Estimation of credibility and model parameters exploits the results of the previous iteration of the validation process, thereby avoiding re-computation. As we will show experimentally, this does not only increase inference efficiency, but also yields a better approximation compared to random estimation.

Our *iCRF* algorithm implements the third step of the validation process introduced in §2.3, i.e., the inference of the implications of user input on the probabilistic credibility model. In the z-th iteration of the validation process, reasoning is based on the probabilistic fact database of the previous iteration and the user input that has been received in the z-th iteration. That is, if c is the claim validated in the z-th iteration, we rely on the probabilistic fact database $Q_{z-1} = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P_{z-1} \rangle$, with $\mathcal{C}^U_{z-1}$ and $\mathcal{C}^L_{z-1}$ being the sets of unlabelled and labelled claims, respectively, as indicated by P_{z-1}. Then, these sets are updated, $\mathcal{C}^U_z = \mathcal{C}^U_{z-1} \setminus \{c\}$ and $\mathcal{C}^L_z = \mathcal{C}^L_{z-1} \cup \{c\}$, and inference returns a new probabilistic fact database $Q_z = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P_z \rangle$.

In each iteration of the validation process, our *iCRF* algorithm adopts the Expectation-Maximization (EM) principle for inference. This choice is motivated by EM's fast convergence, computationally efficiency, and particular usefulness when the likelihood is an exponential function (i.e., maximising log-likelihood becomes maximising a linear function). Specifically, we infer the values of the variables for unlabelled claims $\mathcal{C}^U$ through a configuration of C^U and learn the weight parameters W. By relying on an EM-based approach, we can further naturally integrate user input on the credibility of specific claims. This is a major advantage compared to approaches based on gradient-descent [47] that optimise model parameters, but do not enable the integration of user input and constraints (e.g., on opposing claims).

Inference alternates between an *Expectation* (E-step) and a *Maximization* (M-step), until convergence. EM-based inference is conducted in each iteration of the validation process, while each EM iteration updates the model parameters W. Hence, in the z-th iteration of validation, we obtain sequences $W^0_z, W^1_z, \ldots, W^l_z$ and $P^0_z, P^1_z, \ldots, P^l_z$ of model parameters and credibility probabilities.

E-step: We estimate the credibility probabilities from the current parameter values. The first E-step of the z-th iteration of the validation process is based on parameters W^0_z, given as input from the previous iteration of the validation process, i.e., $W^0_z = W^{l_{z-1}}_{z-1}$, with l_{z-1} as the number of EM iterations in the $z-1$-th iteration of the validation process. In the l-th E-step of the z-th step of the validation process, credibility probabilities are computed as follows:

(1) A sequence of samples Ω^l_z is obtained by Gibbs sampling according to the conditional probability distribution:

$$\begin{aligned} q^l_z(C^U_z) &= Pr(C^U_z \mid C^L_z, D, S; W^l_z) \\ &\propto \prod_{\pi=\{c,d,s\}\in\Pi} Pr^{l-1}_z(c) \times \phi(o(c), d, s; W^l_z). \end{aligned} \tag{6}$$

We incorporate non-equality constraints (Eq. 3) into Gibbs sampling using an idea similar to [61], which, based on matrix factorisation, embeds constraints as factorised functions into the Markov chain Monte Carlo process. Note that Ω^l_z is a sequence, as any configuration of C^U can appear multiple times. We weight the influence of causal interactions (i.e., cliques) by the credibility of their contained claims, so that user input is propagated via mutual interactions between the cliques.

(2) The probability for each claim $c \in \mathcal{C}^U$ without user input is determined by the ratio of Gibbs samples in which c is credible:

$$Pr^l_z(c) = \frac{\sum_{\omega \in \Omega^t_z} \omega(c)}{|\Omega^t_z|}. \quad (7)$$

For all other claims $c \in \mathcal{C}^L$, the probability is fixed by the user input: We set $Pr^l_z(c) = 1$, if the user confirms a claim, and $Pr^l_z(c) = 0$ otherwise.

M-step: We compute the new parameter values by maximising the expectation of log-likelihoods as a weighted average of the probability distribution of current label estimates. That is, in the l-th M-step of the z-th step of the validation process, we have:

$$W^{l+1}_z = \arg\max_{W'} \sum_{C^U} q^l_z(C^U_z) \log Pr(C^L_z, C^U_z | D, S; W') \quad (8)$$

This step is realised by a L2-regularized Trust Region Newton Method [45], suited for large-scale data, where critical information is often sparse (many zero-valued features).

Proposition 1. iCRF *runs in linear time in the size of the dataset.*

Proof. The E-step is implemented by Gibbs sampling, which takes linear time [19, 37] in the number of claims. The M-step is implemented by the Trust Region Newton Method, which also takes linear time in the dataset size [45] . □

3.3 Instantiation of a Grounding

Once the user input of the z-th iteration of the validation process has been incorporated, a grounding is instantiated. This corresponds to the fourth step of the validation process in §2.3, i.e., deciding which claims are deemed credible. Since claims are not independent, we take the configuration with maximal joint probability:

$$g_z(c) = \begin{cases} 1 & \text{if } (c \in \mathcal{C}^L_z) \vee \\ & (o(c) = 1 \wedge o = \arg\max_{C^U_z} Pr(C^U_z \mid C^L_z, D, S; W_z)) \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

However, solving this equation is similar to solving a Boolean satisfiability problem. Thus, we simply leverage the most recent Gibbs sampling result Ω^*_z, obtained during EM, for instantiation. This is defined by a function *decide* as follows:

$$\begin{aligned} g_z(c) &= decide(c, \Omega^*_z) \\ &= \begin{cases} 1 & \text{if } (c \in \mathcal{C}^L_z) \vee \\ & (o(c) = 1 \ \wedge o = \arg\max_{C^U_z} |\{\omega \in \Omega^*_z \mid C^U_z = \omega\}|) \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (10)$$

Consider a set of claims $\mathcal{C} = \{c_1, c_2, c_3\}$ and assume that the last Gibbs sampling comprised three configurations, $\omega_1 = [1, 1, 0]$, $\omega_2 = [1, 0, 0]$, $\omega_3 = [1, 1, 0]$, where the i-th vector element denotes the credibility of claim c_i. Instantiation will return $[1, 1, 0]$ as this configuration appears most often, so that its probability is maximal.

4. USER GUIDANCE

Having discussed (i) inference based on user input and (ii) instantiation of a grounding, we turn to strategies to guide a user in the validation. This corresponds to the first step of the validation process presented in §2.3, i.e., the selection of a claim for validation. We first define a measure of uncertainty for a probabilistic fact database (§4.1). Then, two selection strategies are introduced (§4.2 and §4.3), before they are combined in a hybrid approach (§4.4).

4.1 Uncertainty Measurement

The model of a probabilistic fact database, as constructed above, enables us to quantify the uncertainty related to credibility inference in order to guide a user in the validation process. Let $Q = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P \rangle$ be a probabilistic fact database. Recall that P assigns to each claim $c \in \mathcal{C}$ the probability $P(c)$ of it being credible, while C is the categorical random variable that captures credibility configurations over all claims. We quantify the overall uncertainty of the database by the Shannon entropy over a set of claims:

$$H_C(Q) = -\sum_{C} Pr(C; W) \log Pr(C; W) \quad (11)$$

In our iCRF model, it can be computed exactly by [58, 57]:

$$H_C(Q) = \Phi(W) - \mathbb{E}_W[t(C)]^T W \quad (12)$$

where $\Phi(W) = \sum_C \prod_\pi \phi(o, d, s; W)$ is called the partition function and $\mathbb{E}_W[t(C)] = \nabla\Phi(W)$. Since our model is an acyclic graph with no self statistics, the partition function is computed exactly using Ising methods [57], which run in polynomial time.

We can further scale-up uncertainty computation by approximating the entropy in linear time, as follows:

$$H_C(Q) = -\sum_{c \in \mathcal{C}} [Pr(c) \log Pr(c) + (1 - Pr(c)) \log(1 - Pr(c))] \quad (13)$$

where the claim probabilities are obtained after each EM iteration (i.e., Eq. 7 for unlabelled claims, or directly by the user input for labelled claims). However, this approximation neglects the mutual dependencies between claims.

4.2 Information-driven User Guidance

A first heuristic to guide the selection of claims for validation aims at the maximal reduction in uncertainty under the assumption of trustworthy sources. It exploits the benefit of validating a claim using the notion of information gain from information theory [59].

To capture the impact of user input on a claim c, we define a conditional variant of the entropy measure introduced earlier. It measures the expected entropy of the database under specific validation input:

$$H_C(Q \mid c) = Pr(c) \times H_C(Q^+) + (1 - Pr(c)) \times H_C(Q^-) \quad (14)$$

where $Q^+ = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P^+ \rangle$ and $Q^- = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P^- \rangle$ are inferred from $Q = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P \rangle$ by iCRF (§3.2), under input that confirms the claim, $P^+(c) = 1$, or labels it as non-credible, $P^-(c) = 0$.

To take a decision on which claim to select, we assess the expected difference in uncertainty before and after incorporating input for a claim. The respective change in entropy is the information gain that quantifies the potential benefit of knowing the true value of an unknown variable [59], i.e., the credibility value in our case:

$$IG_C(c) = H_C(Q) - H_C(Q \mid c). \quad (15)$$

Using this notion, we chose the claim that is expected to maximally reduce the uncertainty of the probabilistic fact database. This yields a selection function for information-driven user guidance:

$$select_C(\mathcal{C}) = \arg\max_{c \in \mathcal{C}} \ IG_C(c) \quad (16)$$

Note that we do not need to rank the opposing claim $\neg c$ of a claim c, as their conditional entropies in Eq. 14 will be equivalent.

4.3 Source-driven User Guidance

User guidance as introduced above assumes that sources are trustworthy—an assumption that is often violated in practice. To tackle this issue, we model source trustworthiness by explicitly aggregating over all claims made by a source. More precisely, the likelihood that a source is trustworthy is measured as the fraction of its claims that are considered credible. The latter is derived from the grounding g_z instantiated in the last, the z-th, EM iteration:

$$Pr(s) = \frac{\sum_{c \in \mathcal{C}_s} g_z(c)}{|\mathcal{C}_s|} \tag{17}$$

where $\mathcal{C}_s = \{c \in \mathcal{C} \mid (c, s) \in \Pi\}$ is the set of claims connected to s in the CRF model. Then, the uncertainty of source trustworthiness values is defined as:

$$H_S(Q) = -\sum_{s \in \mathcal{S}} [Pr(s) \log Pr(s) + (1 - Pr(s)) \log(1 - Pr(s))] \tag{18}$$

The conditional entropy when a claim c is validated is:

$$H_S(Q|c) = Pr(c) \times H_S(Q^+) + (1 - Pr(c)) \times H_S(Q^-) \tag{19}$$

where, as detailed above, Q^+ and Q^- are inferred from Q by iCRF under user input that confirms or disproves the claim, i.e., setting $P^+(c) = 1$ for Q^+, or $P^-(c) = 0$ for Q^-, respectively.

As for the first heuristic, we further capture the information gain as the difference in entropy and, based thereon, define the selection function for source-driven user guidance:

$$IG_S(c) = H_S(Q) - H_S(Q|c) \tag{20}$$

$$select_S(\mathcal{C}) = \arg\max_{c \in \mathcal{C}} IG_S(c) \tag{21}$$

Again, we do not need to rank opposing claims.

4.4 Hybrid User Guidance

There is a trade-off between the information-driven and the source-driven strategy for user guidance. Focusing solely on the former may lead to contamination of the claims from trustworthy sources by unreliable sources. An excessively source-driven approach, in turn, may increase the overall user efforts significantly. Thus, we propose a dynamic weighting procedure that to choose among the two strategies. This choice is influenced by two aspects:

Ratio of untrustworthy sources. If there is a high number of unreliable sources, the source-driven strategy is preferred. With little user input, detection of unreliable sources is difficult, though, so that the information-driven strategy is favoured in the beginning.

Error rate. The grounding g_i captures which claims are deemed credible in the i-th iteration of the validation process. If g_i turns out to be mostly incorrect, we have evidence of unreliable sources and favour the source-driven strategy.

Initially, with little user input, we choose the strategy mainly based on the error rate of the grounding. At later stages of the validation process, the number of inferred unreliable sources becomes the dominant factor. The above idea is formalised based on the ratio of unreliable sources in the i-th iteration of the validation process, which is $r_i = (|\{s \in \mathcal{S} \mid Pr(s) < 0.5\}|)/(|\mathcal{S}|)$. The error rate of the grounding is computed by comparing the user input for claim c in the i-th iteration with the credibility value assigned to c in g_{i-1}, i.e., in the previous iteration. Here, we leverage the probability $P_{i-1}(c)$ of the probabilistic fact database $Q_{i-1} = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P_{i-1} \rangle$, of the previous iteration. The error rate is computed as:

$$\epsilon_i = \begin{cases} 1 - Pr_{i-1}(c) & g_{i-1}(c) = 1 \\ Pr_{i-1}(c) & \text{otherwise} \end{cases} \tag{22}$$

Algorithm 1: Validation process for fact checking

```
input  : sets of sources $\mathcal{S}$, documents $\mathcal{D}$, and claims $\mathcal{C}$,
         $\mathcal{C}_s \subseteq \mathcal{C}$ being claims originating from a source $s \in \mathcal{S}$,
         a validation goal $\Delta$, and a user effort budget $b$.
output : the grounding $g$.
1  $\mathcal{C}^U \leftarrow \mathcal{C}; \mathcal{C}^L \leftarrow \emptyset$;
2  $(P_0, \Omega_0^*) \leftarrow iCRF(\mathcal{S}, \mathcal{D}, \mathcal{C}, (c \mapsto 0.5, c \in \mathcal{C}))$;
3  $g_0 \leftarrow (c \mapsto decide(c, \Omega_0^*), c \in \mathcal{C})$;
4  $z_0 \leftarrow 0$;
5  $i \leftarrow 1$;
6  while not $\Delta \wedge i < b$ do
       // (1) Select a claim to validate
7      $x \leftarrow random(0, 1)$;
       // Source-driven or information-driven strategy?
8      if $x < z_{i-1}$ then $c \leftarrow select_S(\mathcal{C}^U)$ ;
9      else $c \leftarrow select_C(\mathcal{C}^U)$ ;
       // (2) Elicit user input
10     Elicit user input $v \in \{0, 1\}$ on $c$;
11     $\mathcal{C}^U \leftarrow \mathcal{C}^U \setminus \{c\}; \mathcal{C}^L \leftarrow \mathcal{C}^L \cup \{c\}$ ;
       // Calculate error rate $\epsilon_i$
12     if $g_{i-1}(c) = 1$ then $\epsilon_i = 1 - P_{i-1}(c)$ ;
13     else $\epsilon_i = P_{i-1}(c)$ ;
       // (3) Infer implications of user input
       // Update credibility of validated claim
14     $P \leftarrow (c \mapsto v \wedge c' \mapsto P_{i-1}(c'), c' \in \mathcal{C}, c' \neq c)$;
       // Conduct inference
15     $(P_i, \Omega_i^*) \leftarrow iCRF(\mathcal{S}, \mathcal{D}, \mathcal{C}, P)$ ;
       // (4) Decide on grounding
       // Instantiate grounding based on samples of last
          iCRF
16     $g_i \leftarrow (c \mapsto decide(c, \Omega_i^*), c \in \mathcal{C})$;
       // Calculate ratio of unreliable sources
17     $r_i = \frac{1}{|\mathcal{S}|} \left| \left\{ s \in \mathcal{S} \mid \frac{\sum_{c \in \mathcal{C}_s} g_i(c)}{|\mathcal{C}_s|} < 0.5 \right\} \right|$ ;
       // Calculate score to choose selection strategy
18     $z_i = 1 - e^{-\left(\epsilon_i \left(1 - \frac{i}{|\mathcal{C}|}\right) + r_i \frac{i}{|\mathcal{C}|}\right)}$;
19     $i \leftarrow i + 1$;
20 return $g_{i-1}$
```

Using the ratio of unreliable sources r_i and the error rate ϵ_i, a we define a score for choosing the source-driven strategy:

$$z_i = 1 - e^{-(\epsilon_i (1 - h_i) + r_i h_i)} \tag{23}$$

where $h_i = i/|\mathcal{C}|$ is the ratio of user input. This score mediates the trade-off between the error rate ϵ_i and the ratio of untrustworthy sources r_i by the ratio of user input h_i. When the ratio h_i is small, the ratio of untrustworthy sources has less influence and the error rate is the dominant factor. When the ratio h_i is large, the ratio of unreliable sources becomes a more dominant factor.

5. COMPLETE VALIDATION PROCESS

Combining the techniques for credibility inference and instantiation of a grounding (§3) with those for user guidance (§4), we define a comprehensive validation process (§5.1). We further outline how robustness against erroneous user input is achieved (§5.2).

5.1 The Algorithm

Our complete validation process for fact checking is defined in Alg. 1. It instantiates the general validation process outlined in §2.3 to address the problem of effort minimisation (Problem 1). As long as the validation goal is not reached and the user effort budget has not been exhausted (line 6), selection of the claim for which user input shall be sought is done either by the source-driven or the information-driven strategy. The choice between strategies is taken by comparing factor z_{i-1} to a random number (line 8), which implements a roulette wheel selection. The second step (lines 10-13) elicits user input for the selected claim and computes the error rate. The third step incorporates the user input in the probabilistic model (line 14) and then conducts credibility inference by means of our iCRF algorithm (line 15). This yields a new probabilistic model P_i, along with the Gibbs sampling result Ω_i^* of the last E-step. Based thereon, in a fourth step, we decide on the new grounding g_i capturing the facts that are considered credible (line 16). The ratio of unreliable sources r_i is calculated to compute score z_i (lines 17-18), used in the next iteration to choose between the selection strategies.

Proposition 2. *An iteration of Alg. 1 (lines 6-19) runs in linear time in the size of the dataset.*

Proof. The time complexity of the iteration of Alg. 1 is dominated by the *iCRF* algorithm, which infers the implications of new user input. Yet, *iCRF* runs in linear time in the dataset size (Prop. 1). □

Applying Alg. 1 in practice, the computation of the information gain for the information-driven or source-driven selection strategy becomes a performance bottleneck. Therefore, we consider two optimisations for this step:

- *Parallelisation:* The computation of information gain for different claims is independent and thus done in parallel.
- *Graph partitioning:* Not all sources share the same claims and not all claims stem from a single source. Hence, as a preprocessing step before seeking user input, the graph representation of the CRF model can be decomposed into its connected components [39]. The resulting smaller CRF models can then be handled more efficiently.

5.2 Robustness Against User Errors

When validating claims, a user may make mistakes, not because of a lack of knowledge, but as a result of the interactions with a validation system [56]. Assuming that a user is confronted with the current inferred credibility of the claim to validate, along with an assessment of related sources and documents, any decision to deviate from the current most likely credibility assignment is typically taken well-motivated. Common mistakes, thus, are accidental confirmations of a (wrong) inferred credibility value of a claim.

Against this background, we incorporate a lightweight confirmation check, triggered after a fixed number of iterations of the validation process. At some step i, for every claim c that has been validated, a grounding $g^i_{\sim c}$ is constructed, using all information of the probabilistic fact database except the validation of c. Then, the label for claim c in $g^i_{\sim c}$ is compared with the respective user input v. If $g^i_{\sim c}(c) \neq v$, then v is identified as a potential mistake and updated accordingly. Intuitively, this check exploits that additional user input may lead to a different inferred credibility value than the one given earlier directly by the user. As inference is based on a large number of validated claims, instead of a single one, it is considered more trustworthy. We will demonstrate experimentally that this check is highly effective when trying to detect user mistakes.

6. METHODS FOR EFFORT REDUCTION

Based on the validation process introduced so far, this section presents methods to further reduce the required user effort. Detecting convergence of our probabilistic model, we discuss when to terminate validation (§6.1). Reducing set-up costs of a user, we then target batching of claims (§6.2).

6.1 Early Termination

In practice, we can improve efficiency by terminating the validation process upon convergence of the results. Below, we define several criteria that indicate such convergence and, therefore, may be employed as additional termination criteria.

Uncertainty reduction rate. A first indicator is the effect of user input in terms of uncertainty reduction. After each iteration in Alg. 1, the probabilistic fact database Q_i becomes Q_{i+1}. The rate of uncertainty reduction is measured as $(H_C(Q_i) - H_C(Q_{i+1}))/H_C(Q_i)$. The rate approaches zero upon convergence, so that validation is stopped once the rate falls below a threshold.

The amount of changes. Instead of considering the probability values of all claims, this indicator incorporates solely the configuration with the highest likelihood. With g_i and g_{i+1} as the groundings of two iterations of Alg. 1, the amount of change is quantified as $|\{c \in \mathcal{C} \mid g_i(c) \neq g_{i+1}(c)\}|$. If this value becomes negligible, i.e., falls below a threshold over several consecutive iterations, we conclude that the credibility of claims has been determined.

Amount of validated predictions. Another indicator for a high quality model is the ability to instantiate credibility assignments that are matched with user input. Exploiting this idea, in each iteration of Alg. 1, we assess whether the result of inference and the user input are consistent. If this is the case for several consecutive iterations, we conclude that the validation process may be stopped.

Precision improvement rate. A more direct way to assess convergence is to estimate the precision based on k-fold cross validation. Formally, in the i-th iteration of Alg. 1, the set of labelled claims $\mathcal{C}^L$ is divided into k equal-size partitions, $E = E_1 \cup \ldots \cup E_k$. Then, we repeat the following procedure k-times: (1) consider the claims of the j-th partition E_j as non-validated; (2) conduct credibility inference ignoring the user input for claims in E_j and instantiate a grounding g'_j; (3) compare the credibility values for claims in E_j based on g'_j with those given directly by the user: $A_{E_j} = (|\{c \in E_j \mid g'_j(c) = P_i(c)\}|)/|E_j|$. We then take the average of k runs as an overall estimation of the model precision at step i, i.e., $A_i = (\sum_{j=1}^{k} A_{E_j})/k$. This yields a rate $(A_i - A_{i-1})/A_{i-1}$ of precision improvement at step i. This rate shall converge to zero, thereby indicating when to terminate the validation process.

6.2 Batching

Batching of claims reduces the set-up costs of users, i.e., the time needed to familiarise with a particular domain. Moreover, batching enables the definition of large validation tasks, which is beneficial when involving multiple users working in parallel. We thus adapt the approach defined in Alg. 1, so that a set of claims, instead of a single one, is checked per iteration. Below, we show how to lift claim selection to sets, assessing the benefit of their joint validation.

Expected benefit. We measure the information gain of validating claims $\mathcal{B} \subseteq \mathcal{C}$ by the expected uncertainty reduction. With B as the categorical random variable that represents credibility configurations of claims $\mathcal{B}$, the uncertainty conditioned by user input on $\mathcal{B}$ is:

$$H_C(Q \mid B) = \sum_{B} Pr(B) H(Q^B) \tag{24}$$

Here, Q^B denotes the probabilistic fact database constructed after incorporating the given configuration of B. Note that a more complex cost model could be constructed based on validation difficulty (e.g., implied by logical relations between claims) [16]. Yet, this is orthogonal to our work. Using this measure, our validation process incorporates batching of claims by choose the top-k claims with maximal information gain (breaking ties randomly):

$$select_B(\mathcal{C}) = \underset{\mathcal{B} \subseteq \mathcal{C}, |\mathcal{B}|=k}{\arg\max}\; H_C(Q) - H_C(Q \mid B) \tag{25}$$

However, the above optimisation problem is computationally hard, as, in practice, both $|\mathcal{C}|$ and k are large. We therefore resort to an *approximate computation* of the benefit and a *greedy algorithm* for the actual selection.

Approximating the expected benefit. We employ an alternative utility function that combines the individual benefit of each claim with a redundancy penalty that incorporates claim dependencies.

Individual benefit: The expected benefit of a claim c is computed as its information gain $IG_C(c)$ as defined in Eq. 15, which is tractable. Selecting claims one-by-one based solely on their individual benefit, however, may be non-optimal, due to the complex joint distribution of random variables for claims, documents, and sources.

Redundancy penalty: Neglecting the dependencies between the variables in the CRF model may yield redundant validation effort. Therefore, when selecting claims, we aim at low information overlap, which is quantified as the redundancy of a set of claims $\mathcal{B} \subseteq \mathcal{C}$ as:

$$R(\mathcal{B}) = \sum_{c,c' \in \mathcal{B}} IG_C(c) M(c,c') IG_C(c') \tag{26}$$

where $M(c,c') = \frac{1}{Z}|\{s \in S | c \in \mathcal{C}_s \wedge c' \in \mathcal{C}_s\}|$ is a correlation matrix that is based on the number of sources that serve as the origin of both claims c and c' and normalised to the unit interval by $Z = \max_{c,c' \in C} M(c,c')$.

Approximated benefit: The two aforementioned measures are combined to approximate the benefit of validating a set of claims $\mathcal{B} \subseteq \mathcal{C}$. The individual benefit, however, is weighted by the importance of a claim. The idea is that claims stemming from a large group of dependent claims have a high chance to propagate information. To exploit this effect, we define $q(c) = \sum_{c' \in \mathcal{C}} M(c,c') IG_C(c')$ as the importance of claim c. Putting it all together, we employ the following utility function to approximate the benefit of validating $\mathcal{B}$:

$$F(\mathcal{B}) = w \sum_{c \in \mathcal{B}} q(c) IG_C(c) - \sum_{c,c' \in \mathcal{B}} IG_C(c) M(c,c') IG_C(c') \tag{27}$$

where $w \in \mathbb{R}^+$ is a positive weight parameter to balance the terms related to individual benefit and redundancy. Then, our utility function is used to guide the selection of the top-k claims:

$$select_{AB}(\mathcal{C}) = \underset{\mathcal{B} \subseteq \mathcal{C}, |\mathcal{B}|=k}{\arg\max} \; F(\mathcal{B}) \tag{28}$$

As discussed, computation of the utility function F is tractable. However, the above optimisation problem (Eq. 28) is not.

Theorem 1. *Computing the result of $select_{AB}$ is NP-complete.*

Proof. F is a submodular set function. Maximization of such functions is known to be NP-complete [49]. □

Greedy selection. Exploiting the monotonicity and submodularity of the utility function F, we define a greedy algorithm with a performance guarantee of $(1 - 1/e) \approx 0.63$ [49]. We iteratively expand the set of claims in k iterations. In each iteration, we traverse all unlabelled claims to identify the claim c^* to maximise the gain $\Delta(c^*) = F(\mathcal{B}' \cup \{c^*\}) - F(\mathcal{B}')$, where $\mathcal{B}'$ is the set of claims selected in the previous iteration. Note that the gain can be updated incrementally. That is, $\Delta_{i+1}(c) = \Delta_i(c) - 2IG_C(c_i^*)M(c,c_i^*)IG_C(c)$, where c_i^* is the claim chosen in iteration i.

The time and space complexity of this heuristic strategy are $\mathcal{O}(|\mathcal{C}|^2 + k|\mathcal{C}|)$ and $\mathcal{O}(|\mathcal{C}|^2)$, respectively. The quadratic term $|C|^2$ in either complexity stems from the calculation of the correlation matrix $M(.,.)$. The linear term $k|C|$ is explained by k iterations, each requiring consideration of the whole set of claims to select c^*.

7. STREAMING FACT CHECKING

We now lift our approach to a streaming setting. Instead of checking a large set of claims from scratch, we consider a potentially infinite stream of claims to validate.

Algorithm 2: Streaming fact checking

```
input : Probabilistic fact database $Q = \langle \mathcal{S}, \mathcal{D}, \mathcal{C}, P \rangle$ and its CRF
        representation $Pr(C|D,S;W)$,
        A potentially infinite stream of claims $c_1, c_2, \ldots$.
1  while a new non-validated claim $c_t$ arrives do
2      $C_t^U \leftarrow C_{t-1}^U \cup \{c_t\}$ ;
3      if $c_t$ comes with a new document $d_t$ then $D_t \leftarrow D_{t-1} \cup \{d_t\}$ ;
4      else $D_t \leftarrow D_{t-1}$ ;
5      if $c_t$ comes with a new source $s_t$ then $S_t \leftarrow S_{t-1} \cup \{s_t\}$ ;
6      else $S_t \leftarrow S_{t-1}$ ;
7      Receive current model parameters $W$ from Alg. 1 ;
8      Compute $Q_t(W)$ using Eq. 29 ;
9      Compute $W_t$ using Eq. 30 ;
10     Feed new model parameters $W_t$ to Alg. 1 ;
```

Upon the arrival of new documents, sources, and claims, the model structure and its parameters need to be updated. However, evaluating the parameters periodically based on the complete database is not a viable option, as the database grows continuously. Limiting the number of considered claims, in turn, may induce a loss of all claims provided by a source. Since only a (small) subset of documents is observed per source, operating on a subset of claims increases the risk of discarding trustworthy sources and documents.

We therefore propose an online expectation-maximization algorithm that reuses and updates the previous trained parameters, which accelerates convergence in the presence of new data. We operate on one claim at a time, and both the claim and the associated user input are discarded after validation. As such, we can only provide an educated guess on the credibility of the claim at a later stage. However, this is a minor drawback, since, in an online setting, claims are relevant only for a comparatively short interval. How to decide on which claims to discard in a more elaborated manner, is an interesting problem, see [51], yet orthogonal to our work.

In the online setting, we consider an EM algorithm with stochastic approximation to update the likelihood with a new claim c_t, a new source s_t, or a new document d_t, rather than conducting re-computation. Specifically, the update rule is defined as:

$$Q_t(W) = Q_{t-1}(W) + \gamma_t \times \left(\underset{C_t^U | C_t^L, D_t, S_t, W_{t-1}}{\mathbb{E}} [\log Pr(C_t^U, C_t^L, D_t, S_t; W)] - Q_{t-1}(W) \right) \tag{29}$$

where $Q_0(W) = 0$ and the sequence $\gamma_1, \gamma_2, \ldots$ is a decreasing sequence of positive step sizes, i.e. $\lim_{T \to \infty} \sum_{t=1}^{T} \gamma_t = \infty$ and $\lim_{T \to \infty} \sum_{t=1}^{T} \gamma_t^2 < \infty$. In practice, the step-size y_t may be adjusted using line searches to ensure that the likelihood is indeed increased in each iteration [18]. As above, the model parameters W are estimated by maximizing the expectation of the likelihood via the L2-regularized Trust Region Newton Method [45]:

$$W_t = \underset{W}{\arg\max} \, Q_t(W) \tag{30}$$

We realise this idea in Alg. 2. Given a stream of claims $c_1, c_2, \ldots$, the algorithm updates the model variables C_t^U, D_t, S_t (lines 2 to 6). It then performs the stochastic approximation of the parameter estimates (lines 8 to 9). The returned parameters are then fed to Alg. 1 (line 10). Alg. 2 can receive the current model parameters from Alg. 1 (line 7), since both algorithms may run in parallel and influence the parameters of one another. The respective parts of either algorithm are highlighted. That is, user input in Alg. 1 or the arrival of a new claim in Alg. 2 may change the model.

Proposition 3. *Alg. 2 runs in linear time.*

Proof. The update of a new claim is implemented by Trust Region Newton Method, which takes linear time [45] in the dataset size. □

8. EVALUATION

We evaluate our approach experimentally, using real-world datasets. We first discuss the experimental setup (§8.1), before turning to an evaluation of the following aspects of our approach:

- The runtime performance of the presented approach (§8.2).
- The efficacy of the CRF model (§8.3).
- The effectiveness of user guidance (§8.4).
- The robustness against erroneous user input (§8.5).
- The effectiveness of early termination (§8.6).
- The benefits and trade-offs of batching (§8.7).
- The streaming setting of fact checking (§8.8).
- The real-world deployment for human validators (§8.9).

8.1 Experimental Setup

Datasets. We utilise state-of-the-art datasets in fact checking [71]:

- *Wikipedia:* The dataset contains proven hoaxes and fictitious people from Wikipedia [10] with 1955 sources, 3228 documents, and 157 labelled claims. The model has been constructed by taking unique, curated claims from Wikipedia and using them as a query for a search engine to collect Web pages as documents, while the originating domain names indicate the sources. The top-30 retrieved documents are linked to a given claim, except those that originate from wikipedia.org in order to avoid a bias, as described in [54].
- *Healthcare forum:* The dataset contains 291276 claims about side-effects of drugs extracted from 2.8M documents of 15K users on healthboards.com [8]. We consider 529 claims of 48083 documents from 11206 users, which have been labelled by health experts. The model has been constructed using domain-specific rules to extract RDF triples from forum texts, i.e. documents. Each user of the forum is considered as a source. Various pattern mining and data cleaning routines are used to ensure that the resulting set of claims does not contain duplicates, see [48].
- *Snopes:* This dataset [9] originates from the by far most reliable and largest platform for fact checking [63], covering different domains such as news websites, social media, and e-mails. The dataset comprises 80421 documents of 23260 sources that contain 4856 labelled claims. The model has been constructed as described above for the *wikipedia* dataset: A duplicate-free set of curated claims of the Snopes' editors was used to collect Web pages that links to these claims, see [54].

For these datasets, we derive features as follows. If a source is a website, we rely on centrality scores such as PageRank and HITS. If a source is an author, features include personal information (age, gender) and activity logs (number of posts). Language quality of documents is assessed using common linguistic features such as stylistic indicators (e.g., use of modals, inferential conjunction) and affective indicators (e.g., sentiments, thematic words) [52].

We follow common practice [46, 14, 34, 50, 29] and use the ground truth of the datasets to simulate user input. Model parameters are initialised with 0.5, following the maximum entropy principle.

Evaluation measures. In addition to the uncertainty of a probabilistic fact database, see §4, we measure:

User effort (E): the ratio of validated claims $|\mathcal{C}^U|$ and all claims $|\mathcal{C}|$, i.e., $E = |\mathcal{C}^U|/|\mathcal{C}|$.

Precision (P_i): the correctness of the grounding. Let $g^* : \mathcal{C} \rightarrow \{0,1\}$ be the correct assignment of credibility values. Then, we measure precision of grounding g_i in the i-th iteration of the validation process as $P_i = |\{c \in \mathcal{C} \mid g_i(c) = g^*(c)\}|/|\mathcal{C}|$. This definition of precision is different from the one in information retrieval and binary classification [59]. As the user interest is a trusted set of facts, the correctness of obtained facts is evaluated.

Precision improvement (R_i): a normalised version of precision, measuring relative improvements to illustrate the effect of user input. With P_0 as the initial precision, the measure is defined at the i-th iteration by $R_i = P_i - P_0/1 - P_0$.

Experimental environment. Our results have been obtained on an Intel Core i7 system (3.4GHz, 12GB RAM). All except the experiments on early termination (§8.6) ran until the actual termination of the validation process.

8.2 Runtime Performance

We first measures the response time, denoted by Δt, of our approach during one iteration of Alg. 1, i.e., the wait time of a user. This includes the time for inference and claim selection.

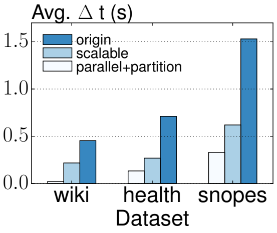

Figure 2: Time vs. dataset

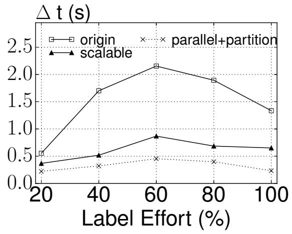

Figure 3: Time vs. effort

Fig. 2 shows the observed response time, averaged over 10 runs, when using the plain algorithm (*origin*), with uncertainty estimation as introduced in §4.1 (*scalable*), and with the computational optimisations of §5.1 (*parallel+partition*). With larger dataset size (*wiki* to *snopes*), the response time increases. However, with computational optimisations, the average response time stays below half a second, which enables immediate user interactions. Fig. 3 further illustrates for the largest dataset, *snopes*, how the response time evolves during validation when averaging the response time over equal bins of relative user effort. The response time peaks between 40% and 60% of user effort, since at these levels, user input enables the most conclusions on credibility values.

8.3 Efficacy of the CRF Model

Next, we assess the estimated probabilities of credibility assignments. Since we use probabilistic information to guide validation, the probabilities should reflect the ground truth, i.e., the true credibility values of claims. For each claim, our model should assign a higher probability to correct credibility values than to incorrect ones. In the experiment, we keep track of the correct assignments (if a claim is correct, we plot $Pr(c = 1)$ and otherwise, we plot $Pr(c = 0)$) and their associated probabilities, while varying the user effort (0%, 20%, 40%).

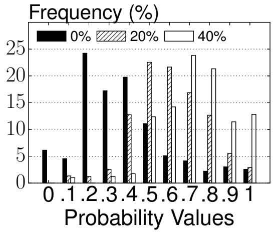

Figure 4: Guidance benefits

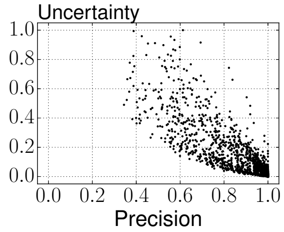

Figure 5: Uncert. vs. prec.

Fig. 4 shows a histogram over all datasets, illustrating how often the probability assigned to a claim falls into a specific bin. Increasing the amount of user effort, the range covering most of the correct credibility values shifts from lower probability bins to higher ones. Even with little user effort (20%), the number of correct assignments with a value ≥ 0.5 is high. This highlights that user input indeed enables a better assessment of the credibility of claims.

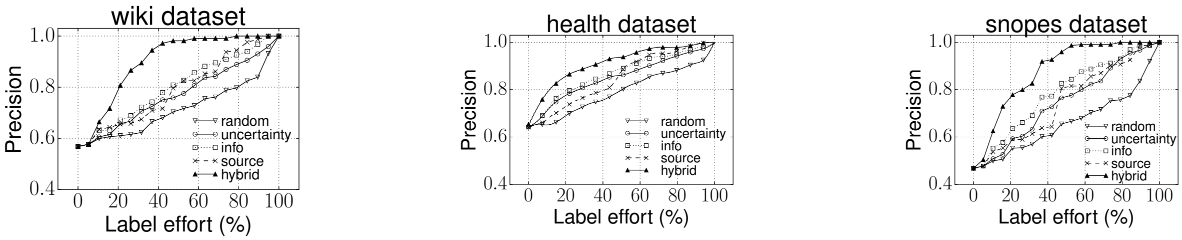

Figure 6: Effectiveness of guiding

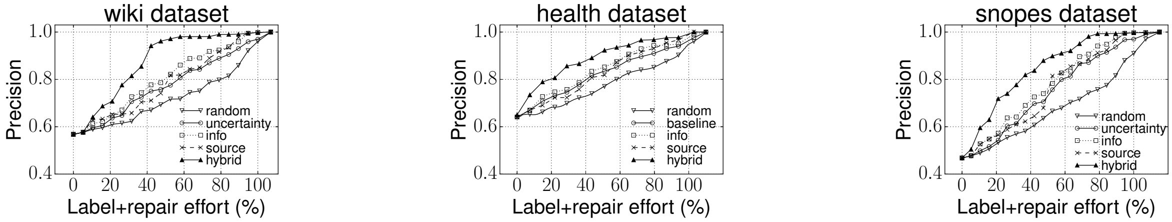

Figure 7: Guiding with erroneous user input

8.4 Effectiveness of User Guidance

Relation between uncertainty and precision. We verify our assumption that the uncertainty of a fact database, see §4, is correlated with the precision of the grounding. In this experiment, the information-driven guidance was applied to all datasets (100 runs each), until precision reaches 1.0. Fig. 5 plots the observed values for precision and normalised uncertainty (i.e., uncertainty divided by the maximum value of the run). There is a strong correlation between both measures (Pearson's coefficient is -0.8523, a highly negative correlation). Hence, uncertainty is indeed a truthful indicator of correctness of the credibility assignments.

Guidance strategies. In this experiment, we mimic the user by the ground-truth, until precision reaches 1.0. We compare our approach (*hybrid*) with four baseline methods: *random*, which selects a claim randomly; *uncertainty*, which selects the most 'problematic' claim, in terms of the entropy of its probability; *info*, which uses the information-driven user guidance only; and *source*, which uses the source-driven user guidance only. Fig. 6 shows the results for all datasets. Our approach (*hybrid*) clearly outperforms baseline techniques. For example, using the *snopes* dataset, our approach leads to a precision value > 0.9 with input on only 31% of the claims, whereas the other methods require validation of at least 67% to reach the same level of precision.

8.5 Robustness Against Erroneous User Input

Detecting erroneous input. We evaluate our approach to detect erroneous input by simulating user mistakes. With a probability p, we transform correct user input into an incorrect assessment. The confirmation check (§5.2) is triggered after each 1% of total validations. Table 1 shows the detected mistakes (%) when increasing parameter p. Across all datasets, the majority of inserted mistakes is detected.

Table 1: Detected mistakes

Dataset	p : probability of mistake			
	0.15	0.20	0.25	0.30
wiki	100	100	96	89
health	100	100	94	86
snopes	100	95	87	79

User guidance with mistakes. We further study the effect of user mistakes on the relation between user effort and precision. Again, the confirmation check is triggered after each 1% of total validations. Upon a detected mistake, the user reconsiders the input, which adds to the invested effort. Fig. 7 illustrates that this implies that more user interactions are required to reach perfect precision. However, the precision curves obtained with our approach are still much better than with other baseline methods.

Effects of missing user input. A user may skip the validation of a claim due to being unsure or preferring to check another claim first. We consider such scenarios by a probability p_m with which a claim is skipped, meaning that the second-best claim is validated. We test p_m ranging from 0.1 to 0.5, while running the validation process until a precision value of 0.7, 0.8, or 0.9 is reached. Fig. 8 shows the saved efforts (%), computed as the relative difference in user effort between the normal process and the one with skipping, needed to reach the respective precision. As expected, skipping at the beginning of the validation process (precision level of 0.7) affects the saved effort, as selecting the second-best candidate leads to worse inference results. Later, this effect becomes smaller.

8.6 Benefits of Early Termination

Using the *snopes* dataset (*wiki* and *health* show similar trends), we evaluate our indicators for early termination of the validation process (see §6.1): the uncertainty reduction rate (*URR*); the amount of changes (*CNG*); the amount of validated predictions (*PRE*); and the precision improvement rate (*PIR*). Fig. 9 plots user effort and precision improvement and, on the secondary Y-axis, the values of the above indicators. Overall, the indicators are aligned with the convergence of the validation process. For example, using the *URR* indicator, validation can be stopped at an *URR* value of 20%. Then, at 40% of user effort, large relative improvements of precision ($> 80\%$) have materialised already.

8.7 Benefits of Batch Validation

Next, we evaluate the benefits of selecting the top-k claims for validation. Here, high values of k lead to larger savings of user set-up costs. Yet, increasing k also implies less accurate estimation of potential benefit, due to our greedy algorithm (§6.2). To explore this trade-off, we capture the costs saved (CS) as a function of k: $CS(k) = 1 - 1/k^{\alpha}$, where α is the rail factor to control the increased cost of validating sets of claims. The chosen function form enables us to capture both linear and non-linear cost models.

Static batch size. When conducting validation with batching, the obtained precision will be lower, since inference is conducted only once the input for the whole batch has been incorporated. We measure this effect by the precision degradation, the relative difference

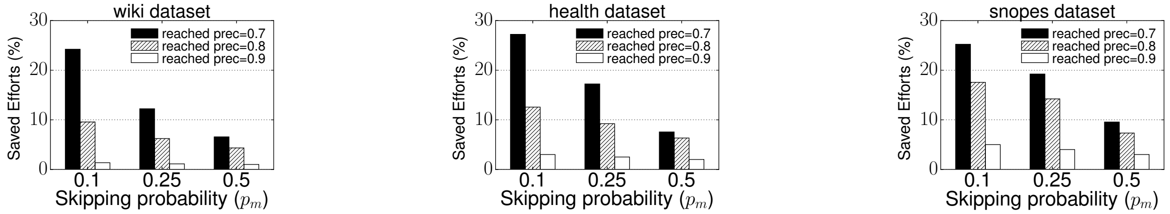

Figure 8: Effects of missing user input

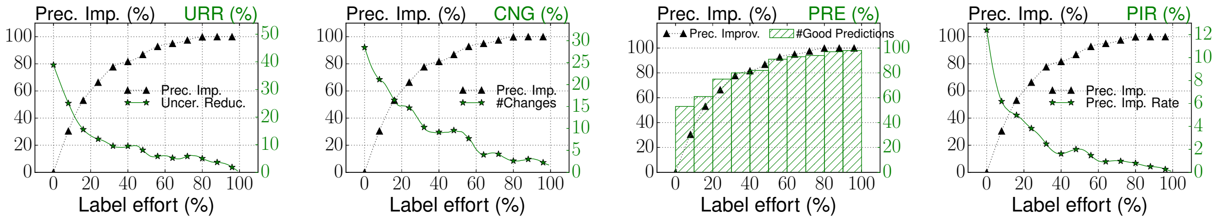

Figure 9: Effectiveness of early termination criteria

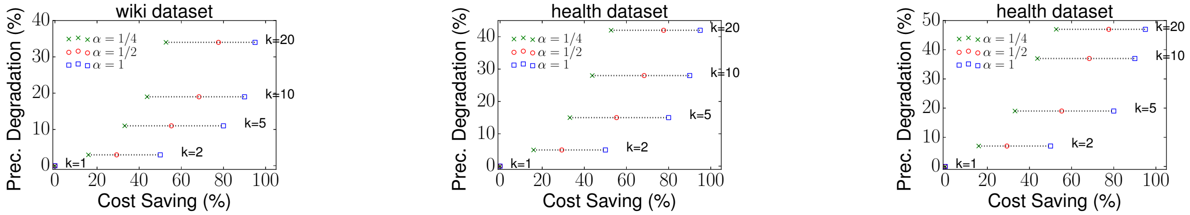

Figure 10: Effects of static batch size

in precision between the validation processes without batching and with batches of size k, varied between one and 20. Fig. 10 plots precision degradation (%) relative to the cost saving (%) using batching under cost models with $\alpha = 0.25, 0.5, 1$. As expected, larger batches lead to lower precision, but increased cost savings. Medium-sized batches ($k = 5, 10$) appear to be beneficial, as they yield potentially large cost savings with a graceful reduction in precision.

Dynamic batch Size. We further explore a dynamic selection of the batch size k, with the goal to maximizes cost savings and precision. We consider different precision thresholds (0.8,0.9) and count the validated claims after each user interaction needed to reach that threshold. For a cost model with $\alpha = 2/3$, Fig. 11 shows box plots of the user effort (%) relative the cost savings (%). Observing the same trade-off as in Fig. 10, the specific results suggest how to choose k dynamically: Initially, a small k shall be used, which is increased once a sufficient amount of claims has been validated.

8.8 Streaming Fact Checking

Update time. We measure the response time during one iteration of Alg. 2, i.e., the update time of the model when a new claim arrives. We run the update process from 0% to 100% of claims in the order of their posting time, for each dataset. The average update time for the *wiki*, *health*, and *snopes* datasets are 0.34s, 0.61s, and 1.22s respectively. As such, the response times turn out to be similar to those of Alg. 1, as implied by Prop. 2 and Prop. 3.

Preservation of validation sequence. As explained in §7, the algorithms for streaming fact checking (Alg. 2) and validation (Alg. 1) run in parallel and update the model parameters. This leads to the question of how to interleave both algorithms: Validating claims early may not be beneficial as later arriving claims help in user guidance. To answer this question, we compare the validation sequences between the offline setting and the streaming setting as follows. We run the streaming algorithm from 0% to 100% of claims in the order of their posting time, and periodically invoke the validation process, where a claim is selected from the existing claims for validation (*hybrid* strategy, current model parameters provided by the streaming algorithm). We record the validation sequence and compare it with the offline setting using Kendall's τ_β rank correlation coefficient [12]. It ranges from -1 (reverse order) to 1 (same order), quantifying the similarity of the ranking in two validation sequences.

Table 2 presents the result when varying the validation period from 5% to 30% (e.g., validation is invoked after every 5% of new claims arrive). Increasing this period, the validation sequence of streaming fact checking becomes more similar to the static setting.

8.9 Real-world Deployment

Finally, we investigate practical issues when deploying our validation framework. A challenge for such an evaluation is that it is difficult to find experts that are knowledgeable in the domains covered by the annotated datasets. Therefore, we consider a setting that features supporting information for the validation. To derive this supporting information, we queried the Google search engine with the text of each claim and extracted the first ten search results as a list of documents. The list of documents is then shuffled for each validation task to avoid biases by the search engine or the user. Due to budget constraints, we selected 50 claims randomly for each dataset. Then, we considered two different types of users:

Experts (E): We implemented a validation interface for expert users, which records the time spent on validation and computes the average accuracy by comparing the answers with the ground truth. We asked three senior computer scientists to complete the validation tasks, with the option to pause between handling different claims.

Crowd workers (C): While it is not the primary use case for our work, crowdsourcing enables scaling of manual validation tasks with the risk of lower result quality due to different levels of worker reliability [33, 30, 32, 32]. We used FigureEight [11] and its web templates to deploy our validation tasks. We prepared a budget

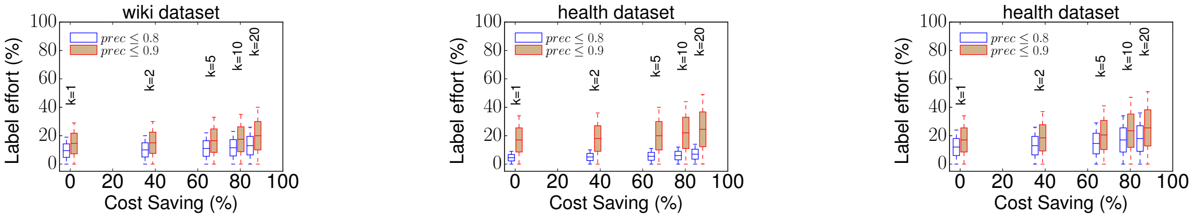

Figure 11: Effects of dynamic batch size

of 1500 HITs (Human Intelligence Tasks) in total with a financial incentive of 0.1$/HIT. We recorded the time spent on validation and computed the consensus of the answers among crowd workers using existing algorithms that include an evaluation of worker reliability [33]. The consensus answer is then compared to the ground truth.

Table 2: Preservation of validation sequence (Kendall's τ_β)

Dataset	validation period			
	5%	10%	20%	30%
wiki	0.23	0.46	0.78	0.84
health	0.19	0.42	0.71	0.78
snopes	0.12	0.38	0.59	0.67

Table 3: Avg. time and accuracy of experts and crowd workers

Dataset	Exp. time	Cro. time	Exp. acc.	Cro. acc.
wiki	268s	186s	0.99	0.88
health	1579s	561s	0.94	0.83
snopes	559s	336s	0.96	0.85

Table 3 summarises the obtained results. Experts validate claims more accurately than crowd workers, but take more time to complete. Moreover, the system also reports that the experts do not validate all the claims in one shoot; the validation process spanned 3-7 days. Note that in our setting, experts and crowd workers already had supporting information in place. Without it, they would have to retrieve such information on their own, which may further increase the validation time. The trade-offs illustrated in Table 3, however, point to the potential benefit of combining the input of experts and crowd workers to achieve efficient, yet accurate fact checking.

9. RELATED WORK

Truth finding on the Web. Given a set of claims of multiple sources, the truth finding (aka fact checking) problem is to determine the truth values of each claim [23]. Existing work in this space also considers mutual reinforcing relations between sources and claims, e.g., by Bayesian models [72], maximum likelihood estimation [64], and latent credibility analysis [53]. However, these techniques neglect posterior knowledge on user input and rely on domain-specific information about sources and data, such as the dependencies between sources and temporal data evolution [23]. The fact checking literature, however, focuses on the classification of claims by credibility, based on a fixed training data. This can be seen as the starting point for our work: We put an expert user in the loop to clean the results obtained by automated classification. Our guidance strategies therefore complement the literature on classifying claims in identifying which potential errors of a classifier are most beneficial to validate by an expert user. At the same time, our approach can also support an expert user in building up a fact database from scratch, in a pay-as-you-go manner. Moreover, our approach goes beyond recent work on offline fact checking, e.g., [54], by including a streaming process to incorporate new claims on-the-fly.

Truth finding is also known as *knowledge verification* [42] and *credibility analysis* [47]. Existing automatic techniques mostly look at features of data, such as number of relevant articles, keywords, and popularity, which are noisy and can be easily dominated by information cascades [42]. Again, posterior knowledge on user input cannot be incorporated. Also, approaches based on gradient-descent [47, 20, 24, 55] only optimise model parameters, but neglect external probability constraints. Fact extraction may be performed by diverse data representations, e.g., knowledge bases [22], web tables [17], semi-structured data [26, 62], or free text [15]. Other work uses co-occurrence information and evidential logs [42, 43], but is limited to quantitative information such as identifying unpopular facts based on the number of mentions [42]. Our work is orthogonal to all the above mentioned. By relying on an abstract data representation, our model is not specific to a particular domain. Our principles of user guidance can further be adapted for many of the above techniques, exploiting its generic notion of uncertainty.

User guidance. Guiding users has been studied in data integration, data repair, crowdsourcing, and recommender systems [36, 67, 44, 70, 69, 68, 65, 31, 21]. Most approaches rely on decision theoretic frameworks to rank candidate data for validation. Despite some similarities in the applied models, however, our approach differs from these approaches in several ways. Unlike existing work that focuses on structured data that is deterministic and traceable, we cope with Web data that is unreliable and potentially non-deterministic. Also, instead of relying on two main sources of information (data and data provider), we incorporate individual features as well as direct and indirect relations between data types (sources, documents, claims).

Our setting is also different from active learning, as we do not require any training data for a user to begin the validation process. Moreover, we incrementally incorporate user input without devising a model from scratch upon receiving new labels. However, stopping criteria for feedback processes have been proposed in active learning, e.g. using held-out labels [46] and performance estimation [40]. Yet, these methods are applicable only for specific classifiers and do not incorporate human factors. Using our probabilistic model, we have been able to propose several criteria for early termination that turned out to be effective in our experimental evaluation.

Moreover, we focus on reducing manual effort, assuming that there is a notion of truth. Yet, user input may be uncertain or subjective [13, 35]. While we consider the integration of such feedback to be future work, we see two scenarios with different implications. First, if claims are validated by a single biased expert [60], the grounding function is shifted to the expert belief. This angle can be extended to recommender systems, which recommend the most belief-compatible claim for a user. Second, if claims are validated by multiple biased experts, differences in their belief suddenly have an impact. Finding a common ground then requires negotiation and conflict resolution mechanisms [27].

10. CONCLUSIONS

In this paper, we proposed an approach to overcome the limitations of existing methods for automatic and manual fact checking. We introduced an iterative validation process, which, based on a probabilistic model, selects claims for which validation is most beneficial, infers the implications of user input, and enables grounding of the credibility values of claims at any time. We further proposed methods for early termination of validation, efficient batching strategies, and a streaming version of our framework. Our experiments showed that our approach outperforms respective baseline methods, saving up to a half of user effort when striving for 90% precision.

11. REFERENCES

[1] Dbpedia. http://www.dbpedia.org.

[2] Earthsky, example claim. http://earthsky.org/human-world/does-eating-turkey-make-you-sleepy.

[3] Freebase. http://www.freebase.com.

[4] Kidshealth, example claim. http://kidshealth.org/en/kids/turkey-sleepy.html.

[5] Snopes, example claim assessment. http://www.snopes.com/food/ingredient/turkey.asp.

[6] Webmd, example claim. http://www.webmd.com/food-recipes/the-truth-about-tryptophan.

[7] Yago. http://www.mpi-inf.mpg.de/yago.

[8] http://resources.mpi-inf.mpg.de/impact/peopleondrugs/data.tar.gz, 2017.

[9] http://resources.mpi-inf.mpg.de/impact/web_credibility_analysis/Snopes.tar.gz, 2017.

[10] http://resources.mpi-inf.mpg.de/impact/web_credibility_analysis/Wikipedia.tar.gz, 2017.

[11] https://www.figure-eight.com, 2018.

[12] A. Agresti. *Analysis of ordinal categorical data*. John Wiley & Sons, 2010.

[13] Y. Amsterdamer, Y. Grossman, T. Milo, and P. Senellart. Crowd mining. In *SIGMOD*, pages 241–252, 2013.

[14] A. Arasu, M. Götz, and R. Kaushik. On active learning of record matching packages. In *SIGMOD*, pages 783–794, 2010.

[15] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, pages 2670–2676, 2007.

[16] S. Basu Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDBJ*, 24(4):467–491, 2015.

[17] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: exploring the power of tables on the web. In *VLDB*, pages 538–549, 2008.

[18] O. Cappé and E. Moulines. On-line expectation–maximization algorithm for latent data models. *J R Stat Soc Series B Stat Methodol*, 71(3):593–613, 2009.

[19] G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[20] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li. Pme: projected metric embedding on heterogeneous networks for link prediction. In *KDD*, pages 1177–1186, 2018.

[21] P. T. Cong, N. T. Toan, N. Q. V. Hung, and B. Stantic. Minimizing efforts in reconciling participatory sensing data. In *WIMS*, page 49, 2018.

[22] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610, 2014.

[23] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. In *VLDB*, pages 37–48, 2012.

[24] C. T. Duong, Q. V. H. Nguyen, S. Wang, and B. Stantic. Provenance-based rumor detection. In *ADC*, pages 125–137, 2017.

[25] C. Elkan. Log-linear models and conditional random fields. *Tutorial notes at CIKM*, 8:1–12, 2008.

[26] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall. In *WWW*, pages 100–110, 2004.

[27] W. Gatterbauer, M. Balazinska, N. Khoussainova, and D. Suciu. Believe it or not: adding belief annotations to databases. In *VLDB*, pages 1–12, 2009.

[28] K. S. Hasan and V. Ng. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *EMNLP*, pages 751–762, 2014.

[29] N. Q. Hung, D. C. Thang, N. T. Tam, M. Weidlich, K. Aberer, H. Yin, and X. Zhou. Answer validation for generic crowdsourcing tasks with minimal efforts. *VLDBJ*, 26(6):855–880, 2017.

[30] N. Q. V. Hung, C. T. Duong, N. T. Tam, M. Weidlich, K. Aberer, H. Yin, and X. Zhou. Argument discovery via crowdsourcing. *VLDB J.*, pages 511–535, 2017.

[31] N. Q. V. Hung, N. T. Tam, V. T. Chau, T. K. Wijaya, Z. Miklós, K. Aberer, A. Gal, and M. Weidlich. SMART: A tool for analyzing and reconciling schema matching networks. In *ICDE*, pages 1488–1491, 2015.

[32] N. Q. V. Hung, N. T. Tam, Z. Miklós, and K. Aberer. Reconciling schema matching networks through crowdsourcing. *EAI*, page e2, 2014.

[33] N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer. An evaluation of aggregation techniques in crowdsourcing. In *WISE*, pages 1–15, 2013.

[34] N. Q. V. Hung, D. C. Thang, M. Weidlich, and K. Aberer. Minimizing efforts in validating crowd answers. In *SIGMOD*, pages 999–1014, 2015.

[35] N. Q. V. Hung, K. Zheng, M. Weidlich, B. Zheng, H. Yin, N. T. Tam, and B. Stantic. What-if analysis with conflicting goals: Recommending data ranges for exploration. In *ICDE*, pages 1–12, 2018.

[36] S. R. Jeffery, M. J. Franklin, and A. Y. Halevy. Pay-as-you-go user feedback for dataspace systems. In *SIGMOD*, pages 847–860, 2008.

[37] C.-J. Kim, C. R. Nelson, et al. State-space models with regime switching: classical and gibbs-sampling approaches with applications. *MIT Press Books*, 1, 1999.

[38] D. Knoke, P. J. Burke, and P. Burke. *Log-linear models*, volume 20. Sage, 1980.

[39] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *TIT*, pages 498–519, 2001.

[40] F. Laws and H. Schätze. Stopping criteria for active learning of named entity recognition. In *ICCL*, pages 465–472, 2008.

[41] J. Lehmann, D. Gerber, M. Morsey, and A.-C. N. Ngomo. Defacto-deep fact validation. In *ISWC*, pages 312–327, 2012.

[42] F. Li, X. L. Dong, A. Langen, and Y. Li. Knowledge verification for long-tail verticals. In *VLDB*, pages 1370–1381, 2017.

[43] X. Li, W. Meng, and C. Yu. T-verifier: Verifying truthfulness of fact statements. In *ICDE*, pages 63–74, 2011.

[44] Y. Li, H. Su, U. Demiryurek, B. Zheng, T. He, and C. Shahabi. Pare: A system for personalized route guidance. In *WWW*, pages 637–646, 2017.

[45] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region newton method for logistic regression. *JMLR*, pages 627–650, 2008.

[46] B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. In *VLDB*, pages 125–136, 2014.

[47] S. Mukherjee and G. Weikum. Leveraging joint interactions for credibility analysis in news communities. In *CIKM*, pages 353–362, 2015.

[48] S. Mukherjee, G. Weikum, and C. Danescu-Niculescu-Mizil. People on drugs: credibility of user statements in health communities. In *KDD*, pages 65–74, 2014.

[49] G. L. Nemhauser and L. A. Wolsey. Maximizing submodular set functions: formulations and analysis of algorithms. *North-Holland Mathematics Studies*, pages 279–301, 1981.

[50] Q. V. H. Nguyen, T. T. Nguyen, Z. Miklós, K. Aberer, A. Gal, and M. Weidlich. Pay-as-you-go reconciliation in schema matching networks. In *ICDE*, pages 220–231, 2014.

[51] T. T. Nguyen, C. T. Duong, M. Weidlich, H. Yin, and Q. V. H. Nguyen. Retaining data from streams of social platforms with minimal regret. In *IJCAI*, pages 2850–2856, 2017.

[52] A. Olteanu, S. Peshterliev, X. Liu, and K. Aberer. Web credibility: Features exploration and credibility prediction. In *ECIR*, pages 557–568, 2013.

[53] J. Pasternack and D. Roth. Latent credibility analysis. In *WWW*, pages 1009–1020, 2013.

[54] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *WWW Companion*, pages 1003–1012, 2017.

[55] T. Qian, B. Liu, Q. V. H. Nguyen, and H. Yin. Spatiotemporal representation learning for translation-based poi recommendation. *TOIS*, 37(2):18, 2019.

[56] J. Reason. *Human error*. Cambridge university press, 1990.

[57] M. G. Reyes. Covariance and entropy in markov random fields. In *ITA*, pages 1–6, 2013.

[58] M. G. Reyes and D. L. Neuhoff. Entropy bounds for a markov random subfield. In *ISIT*, pages 309–313, 2009.

[59] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[60] M. Samadi, P. P. Talukdar, M. M. Veloso, and M. Blum. Claimeval: Integrated and flexible framework for claim evaluation using credibility of sources. In *AAAI*, pages 222–228, 2016.

[61] M. Schmidt. Linearly constrained bayesian matrix factorization for blind source separation. In *NIPS*, pages 1624–1632, 2009.

[62] L. H. Tran, Q. V. H. Nguyen, N. H. Do, and Z. Yan. Robust and hierarchical stop discovery in sparse and diverse trajectories. Technical report, EPFL, 2011.

[63] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[64] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher. On truth discovery in social sensing: A maximum likelihood estimation approach. In *IPSN*, pages 233–244, 2012.

[65] W. Wang, H. Yin, Z. Huang, Q. Wang, X. Du, and Q. V. H. Nguyen. Streaming ranking based recommender systems. In *SIGIR*, pages 525–534, 2018.

[66] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu. Toward computational fact-checking. In *VLDB*, pages 589–600, 2014.

[67] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. In *VLDB*, pages 279–289, 2011.

[68] H. Yin, L. Chen, W. Wang, X. Du, N. Q. V. Hung, and X. Zhou. Mobi-sage: A sparse additive generative model for mobile app recommendation. In *ICDE*, pages 75–78, 2017.

[69] H. Yin, Z. Hu, X. Zhou, H. Wang, K. Zheng, N. Q. V. Hung, and S. W. Sadiq. Discovering interpretable geo-social communities for user behavior prediction. In *ICDE*, pages 942–953, 2016.

[70] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and N. Q. V. Hung. Adapting to user interest drift for POI recommendation. *TKDE*, pages 2566–2581, 2016.

[71] G. Zhang and C. Li. Maverick: a system for discovering exceptional facts from knowledge graphs. In *VLDB*, pages 1934–1937, 2018.

[72] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han. A bayesian approach to discovering truth from conflicting sources for data integration. In *VLDB*, pages 550–561, 2012.