LEAPFROG: Adapting Belief Propagation for Knowledge Graph Construction

Anonymous authors

Abstract

Populating a knowledge base (KB) from unstructured information has been a widely studied problem. Capturing complex events and relations is especially challenging. Even more challenging is providing coherent interpretations of uncertain and even contradictory information. In this work, we present a novel probabilistic framework — LEAPFROG, for automated knowledge base construction that maintains alternative probabilistic interpretations for entities, events and relations. To the best of our knowledge, this work is the first attempt at capturing multiple uncertain alternatives. Furthermore, we allow for a domain expert to inject their beliefs and prior knowledge into the system. We show how the expert’s beliefs about the reliability of an information source affect information interpretation.

1. Introduction

Knowledge bases (KB) provide a structured and unified way for representing unstructured information. They hence play a central role in many real-world applications and tools. For instance, web search engines like Google and Bing are typically built over a huge underlying knowledge graph (KG), which is essentially a connected KB. Current KG’s usually capture one interpretation of an event or a relation and do not account for the possibility of alternative interpretations. One reason is because most large-scale KGs like DBPedia [Auer et al., 2007], Freebase [Bollacker et al., 2008], etc have been built over good quality corpora, which usually contain verified facts. However, in a more realistic setting, data comes from various sources which often contain noisy and conflicting pieces of information. We propose a probabilistic graphical framework called LEAPFROG, which allows for capturing alternative interpretations of entities, events and relations, during the construction of a knowledge graph. Having a probabilistic method allows for assigning confidence scores to these alternative interpretations thus providing a ranked list to the end-user. Figure 1 shows an example graph for a disputed event shooting of MH17 generated by LEAPFROG. There is much debate about what exactly caused the downing of the flight MH17 in 2014,
and we observed that most documents support the hypothesis that a *BUK missile* caused it.

![Figure 1: A knowledge graph entry for the conflicting event "shooting of MH17". The event is of *Conflict.Attack* type and is represented using the semantic roles — target/patient, location, instrument. The value of each role shows a probability distribution over the alternatives. For e.g. most documents support *BUK missile* being the instrument of attack.](image)

The goal of this work is not to judge the veracity of an information source but to provide the end user with a multifaceted view of an event. Along with presenting a ranked list of the most probable alternatives, we also provide the provenance/evidence pointers that lend support to the presented ranking. Since automatically constructed knowledge bases often contain incorrect or incomplete information, some have been designed to incorporate feedback from the user; NELL [Mitchell et al., 2018] prompts users to correct facts; Kobren et al. [2017] asks users to augment or correct missing attributes of an entity. Our work differs in that, we provide the user with the flexibility to inject their prior beliefs regarding events and beliefs about reliability information sources. For instance, if a user has 100% trust in news reported by *BBC* but only 20% trust in *RT (Russia Today)*, then LEAPFROG will construct a different KG if the trust was reversed.

We evaluate our proposed approach on the TAC-KBP 2018 data, as part of SM-KBP 2018 workshop ¹ organized by NIST ², under the aegis of DARPA. This paper makes the following contributions:

1. We propose a probabilistic graphical model for populating a knowledge graph in the presence of conflicting and uncertain information. This information includes text and images from documents in several languages. The proposed model provides a ranked list of alternative interpretations for these conflicting events and relations.

2. To the best of our knowledge, we are the first to provide the ability to bias the knowledge representation based on human beliefs (e.g., beliefs about the reliability of information sources).

². https://www.nist.gov/
Figure 2: System overview for LEAPFROG. It takes entity/event/relation mentions along with the domain ontology as inputs and uses belief propagation based engine to output a knowledge graph having multiple interpretations for conflicting information.

3. As a by-product, we perform entity, event and relation co-reference both within and across documents. Another by-product is knowledge completion as the model fills in missing argument slots.

2. LEAPFROG

In this section, we describe LEAPFROG and discuss the underlying probabilistic model governing it. The system overview can be seen in Figure 2. Entity, event and relation mentions are the primary inputs to LEAPFROG. Ontology is an initial input to LEAPFROG to provide domain knowledge. It can also process additional inputs, in the form of cross-modal and cross-lingual entity co-reference. The underlying probabilistic model of LEAPFROG is a belief graph (BG), which is our version of KG. The belief propagation engine performs inference by calculating the marginal probabilities of the KG nodes to update the belief graph. LEAPFROG generates one "mini" belief graph per document as an intermediate step in the incremental BG construction.

2.1 Belief Graph

The BG nodes are Knowledge Elements (KEs) such as entities, events and relations. KEs have properties. For instance, name and entity-type are two properties of an entity, where entity-type can be any one of the following: \{person, weapon, vehicle, commodity, geopolitical, location, facility\}, as defined by the SM-KBP Ontology. Events have event-type like Conflict.Attack as seen in Figure 1 and specific arguments like (Conflict.Attack-Attacker, Conflict.Attack-Target, etc) as its properties. Relations also have a relation-type and

are usually binary, depicting affiliations (OrganizationAffiliation, EmploymentMembership, Physical.OrganizationHeadquarter, etc.), between entities. A natural way to connect KEs to their uncertain properties is through random variables. The uncertainty is reflected in the variable’s probability distribution. To propagate and, hopefully, reduce the uncertainties observed in the data, we construct factor graphs over the BG.

**Factor graph:** Factor graph is a probabilistic graphical model that represents factorization over several variables. It allows for efficient computation of marginal distributions using a sum-product algorithm, belief propagation (BP) [Pearl, 2014], discussed in detail in Section 2.3.

**Open sets:** When a variable has many potential values, we need a method for representing probability distributions without listing all the values that a variable can have. We propose to do it using the OpenNominal distribution type. For example, the name property of an entity $P$ of type Person with $n$ potential values would be:

$$P.\text{name} = \text{Variable(OpenNominal, \{Bill: .6, Bob: .3, *other: .1\}, N = n)}$$

This means that the probability of $P$’s name being, say ”Roger” would be $0.1$. This serves two purposes: a) it keeps the number of potential values from growing too large which would make the inference over the factor graph intractable and b) it doesn’t disregard the possibility of a variable taking a value which isn’t observed yet, which is usually the case in real-world documents.

We follow a two-step approach for BG construction — evidence graph construction using the input observations, and evidence graph grounding to the BG.
2.2 Evidence Graph Construction

The inputs to this module are entity, event and relation mentions $M = \{m_i\}_{i=0}^n$. The entity mentions are mapped to the respective event/relation roles. For instance, consider the event mention E1 describing an event of type `Attack` in Figure 3. Since each event has event-specific arguments, we map these arguments to a fixed set of semantic roles such as Agent, Target, etc., which serves the dual purpose of being consistent as well as making the computation tractable. We acknowledge that mapping event-specific arguments to a fixed set might not capture all aspects of the event, but for most events in our ontology such a mapping wasn’t detrimental. Therefore, the evidence graph for this event mention has a graph node denoting the event type and its semantic roles `Target` and `Instrument`. The entity mention `flight MH17` is mapped to its corresponding node of `Target` and similarly `Buk missile` to the role `Instrument`. The top half of Figure 3 shows the evidence graph for other observed events.

2.3 Evidence Graph Grounding

Evidence is grounded to the Knowledge Elements (KEs) of KG via random variables $V$ as seen in the bottom half of Figure 3. For a given evidence (event/relation), first the respective arguments (entities) are grounded. Our knowledge graph may contain millions of variables and dependencies. We do not use them all at once as it is wasteful and may never converge. Instead, when new information becomes available, we use a combination of general and domain-specific logic to select the relevant variables and dependencies for belief propagation.

**Entity Grounding:** As a first step, candidate KEs are retrieved from the KG based on the argument entity’s `entity-type`. We further filter these candidates using fuzzy string matching. Additionally, we leverage relation mentions for further reduction of the candidate set. Affiliation relations such as ownership are used to disambiguate entities having similar names. For example, the entity `Spanish truck` has `entity-name: truck` and a relation `owned-by: Spain`. Another entity `French truck` has the same `entity-name: truck` but a different owner `owned-by: France`. Thus, using only name based heuristics will fail to disambiguate these entities.

**Event and Relation Grounding:** We use the same first filtration step, as used for entities, for getting the initial candidate set of event/relation KEs based on the observations’ `event-type/relation-type`. Potential candidates are further filtered by selecting those event KEs that have overlapping entity grounding for at least one of the observed event’s respective argument. For example, for the evidence mention `{transport of cash by Spanish truck ...}`, a potential candidate within the same document would have `event-type = Movement.TransportArtifact` and at least one of its argument roles (patient, instrument) matching the respective role’s entity grounding. In this case, the grounding for `cash` entity in the candidate KE should match the grounding for `cash` (patient) in the mention, or grounding for `Spanish truck` should match the grounding for `Spanish truck` (instrument) in the mention. Since a single document is usually centered around limited topics, such a heuristic works well for events within the same document. Across multiple documents, such a heuristic adds irrelevant candidates so we add a more restrictive constraint by having
candidates match at least the patient argument. In the above case, a potential candidate must have event-type = Movement.TransportArtifact and patient grounding similar to the entity grounding of cash.

Relations also use similar heuristics, since they can be treated as events with two arguments. In our domain, relations usually encode ownership affiliations and location hierarchy. For example, Spanish truck has relation-type=GeneralAffiliation.APORA, where owner-entity: Spain and owned-entity: truck. Grounding candidates are filtered based on relation-type and a similar entity grounding for the owned-entity role. Usually in affiliation relations, the uncertainty lies with the owner role. In the same example of Spanish truck and French truck, the uncertainty could lie with the owner of the truck entity.

Most importantly, for both events and relations we remove candidates that were cited by the same source. Source here refers to the person/organization held responsible for a statement, for example, {the Defense Ministry said that ...} or {... statement by the President...}. Since multiple documents often quote the same sources, we avoid re-counting the same piece of information. Additionally, we store the provenance of the mentions in the grounded KEs. This keeps track of the evidence used by the model during entity/event/relation grounding. Once the candidates are selected, we form the factor graph. A new KE is created if the candidate set is empty.

Factors: Dependencies between random variables are captured through factors. Suppose an observer reports seeing a truck but not sure if the truck is Spanish or French. We capture this with an observed entity Obs and two variables corresponding to the observed properties: Obs.name with distribution {truck: 1.0} and Obs.affiliation with distribution {Spanish: 0.5, French: 0.5}. Lets assume that we have two existing grounding candidates Truck1 and Truck2. We also consider the possibility of the observed entity being a new truck - Truck3. Obs.target is the grounding variable for the observation with a uniform prior {Truck1: 0.33, Truck2: 0.33, Truck3: 0.33}. We form a factor that connects these three variables as depicted on Figure 4. The local function of this factor takes three arguments - the instances of Obs.name, Obs.affiliation, Obs.target - and returns 1 if the entity-name and affiliation of the observation match the entity-name and affiliation of the grounding candidate and 0 otherwise. LEAPFROG has several common observation models with pre-built factors and local functions.

Belief Propagation: Belief propagation (BP) is a message-passing algorithm for performing inference over graphical models. It efficiently computes the posteriors of the target variables and the algorithm stops when the prior and the posterior distributions of the target variables converge, as measured by their Kullback-Leibler distance. We explicitly list only the highest posteriors, relegating the lower ones to *other or rejecting them. For example, if the posterior probability of the new truck in the previous example is below a threshold, we do not create the new entity in the KB. Observation target variables serve as a co-reference resolution mechanism: given two observations, we can look at their grounding targets and calculate the probability of an intersection. LEAPFROG additionally performs knowledge completion for event roles as part of this process. If after BP, the model decides that the observation is an alternative interpretation, a union of the observed mention’s arguments and arguments of the existing event KE to which the observation was resolved to, is taken. The update equations [Gormley and Eisner, 2014] can be seen below, where $x_i$ are variables,
LEAPFROG: Adapting Belief Propagation for Knowledge Graph Construction.

![Factor graph for either grounding entity mention O1 to existing KE variables (Truck1: French truck) and (Truck2: Spanish truck) or creating a new KE Truck3. The target variable T is set to have a uniform prior, suggesting all possibilities are equally likely initially.]

Figure 4: Factor graph for either grounding entity mention O1 to existing KE variables (Truck1: French truck) and (Truck2: Spanish truck) or creating a new KE Truck3. The target variable T is set to have a uniform prior, suggesting all possibilities are equally likely initially.

α are the factors, \( N(x) \) are neighboring nodes of \( x \), \( \psi_\alpha \) are the factor potentials, which in our case are the identity local functions, \( \mu_{x\rightarrow y} \) are incoming messages from node \( x \) to node \( y \).

Belief of variable \( x_i \): 
\[
b_i(x_i) = \prod_{\alpha \in N(i)} \mu_{\alpha\rightarrow i}(x_i) \quad (1)
\]

Outgoing message from variable \( x_i \): 
\[
\mu_{\alpha\rightarrow i}(x_i) = \prod_{\alpha \in N(i) \setminus \alpha} \mu_{\alpha\rightarrow i}(x_i) \quad (2)
\]

Factor belief: 
\[
b_\alpha(x_\alpha) = \psi_\alpha(x_\alpha) \prod_{i \in N(\alpha)} \mu_{i\rightarrow \alpha}(x_\alpha[i]) \quad (3)
\]

Factor message: 
\[
\mu_\alpha(x_i) = \sum_{x_\alpha:x_\alpha[i]=x_i} \psi_\alpha(x_\alpha) \prod_{j \in N(\alpha)_\alpha} \mu_{j\rightarrow \alpha}(x_\alpha[i]) \quad (4)
\]

LEAPFROG constructs the KB in an incremental fashion and resolves the observations sequentially. This causes the beliefs to be biased towards interpretations observed early in the process.

3. Information Uncertainty

The above described probabilistic approach allows us to model uncertainties in the information. We address two kinds of uncertainties here: 1) source reliability and, 2) mention uncertainty, but it can be easily extended to capture other kinds of uncertainties as well.

3.1 Source reliability

In realistic situations, information comes from multiple sources where not all sources are equally reliable. Consider the observation Spanish truck, and let’s assume that reliability of this document/source is 85%, which means that 85% of the time this particular source reports correct information and 15% times the source guesses (i.e., samples from a "guess").
distribution which may or may not correspond to reality). To model a reader’s belief in the trustworthiness of a source, we have a reliability factor as shown in Figure 5. Obs.say (V1) captures what the source reported, and Obs.guess (V2) captures the distribution from which the source guesses and Obs.reliability (V3) is a boolean random variable denoting the percent reliability. The default guess distribution is uniform. Based on prior knowledge, a different distribution can be specified for the guess variable. The posterior of Obs (O1) encodes the mixture of reporting reality and guessing.

\[ P(\text{obs.property} = x) = P(\text{say}=x)*P(\text{reliable}) + P(\text{guess}=x)*(1- P(\text{reliable})) \]

Factor f1 captures the relation between the three parts of the observation using the above function.

3.2 Mention uncertainty

We also consider the uncertainty of a mention as reported by the author. For instance, in this observation \{ the Spanish truck could have been used for transporting cars \} the author is hedging about the Spanish truck being the instrument of Movement.Transport event. We consider three possibilities for an event/relation mention, a) the author is confident about the observation and reports it with full certainty, b) author is hedging and c) the observed event/relation contains negation. Similar to Section 3.1, we represent this uncertainty using a factor f2, as seen in Figure 5. Like V2, V4 captures the guess distribution and O2 is the mixture of actual observation and guessing. In Figure 5, Obs.certainty V5 is 0.50 which means the author is only 50% certain of the observed event argument and f2 captures this relation to compute the posterior probability of O2 as follows:

\[ P(\text{obs.property} = x) = P(O1=x)*P(\text{certainty}) + P(\text{guess}=x)*(1- P(\text{certainty})) \]

The boxes on top of the respective entity variables O1,O2 are the posterior distributions after the two inference steps respectively. During entity, event or relation grounding, O1 in Figure 4 can be substituted by O2, whose posteriors now encode these uncertainties. Using such a factor-based method, one can incorporate other types of uncertainties by either
creating sequential factors (as above) or modeling all uncertainties in one complex factor. As the certainty of an observation decreases, the probability tends to concentrate around *other* as seen in Figure 5. This reflects human behavior as well — if a reported observation is highly uncertain, human tendency is to have less belief in what was reported and have an open mind towards other possibilities.

### 3.3 Graph Node Merging and Pruning

We implement a *merging strategy* for merging existing KEs in light of new knowledge. This new knowledge can be external (e.g. cross-lingual co-reference) or acquired by the model in the process of KG construction. Consider two KEs *Spanish truck* (KE$_1$) and *French truck* (KE$_2$). Some later events/relations reveal that these were in fact the same truck and hence need to be merged. However, it is not enough to just assign KE$_1$ = KE$_2$, because these entities could be arguments of other event and relation KEs. Therefore, for every merge we follow these steps:

1. Substitute KE$_2$ with KE$_1$ across all corresponding events and relations where KE$_2$ is the argument.

2. For each updated event and relation KE, run the BP again by selecting relevant candidates from existing KG. This is done as merging entity KEs could lead to the merging of event and relation KEs as well. For example, the two separate event KEs *Transport by Spanish truck* and *Transport by French truck* should also get merged. We leverage the same factor graph model for merging these events/relations as described in Section 2.3.

To keep the BP computation tractable and to avoid probability underflow, we prune the graph nodes by removing instances having very low probability. Currently, we only prune entity nodes as the number of entities is much larger than the number of events and relations. Periodically, all entity nodes are pruned removing interpretations that have probability less than the specified threshold. Period $k$ is a hyperparameter which should be chosen based on the domain or quality of corpus, as having too small a $k$ could run the risk of pruning away possibilities which could be important later. This risk is higher for event and relation arguments, and that is another reason why we focus only on entity nodes.

### 4. Evaluation

In this section, we discuss our evaluation results on the pilot data provided as part of SM-KBP 2018 workshop. We also study the effect of biasing the document sources based on trustworthiness, on the full KG which presents some interesting insights.

#### 4.1 Dataset

The provided documents are in English, Russian and Ukrainian languages and consist of news articles and social media data from Twitter. These documents are centered around three conflicting scenarios: T101 (Crash of Malaysian Air Flight MH17), T102 (Flight of Deposed Ukrainian President Viktor Yanukovych), T103 (Who Started the Shooting at
Maidan?). Table 1 shows the number of root documents, including both text and images within that document, for each topic and language. We use the following LDC \(^4\) provided annotations: entity, event and relation mentions; observation uncertainties (hedging and negation) at both the event and argument level. We do not leverage the provided within-document or cross-document entity/event/relation co-reference, except for the cross-lingual and cross-modal entity co-references. However, only single cross-lingual entity co-reference was provided between one English document and one Russian and/or Ukrainian document. The cross-modal co-reference links entities in images with their corresponding text documents. These inputs can also be extracted using off-the-shelf tools, however we observed those to be very noisy given the lack of training data. Since the purpose of this work is to demonstrate a method for handling alternative interpretations and information uncertainties, we show results and analysis using the provided annotations for avoiding error propagation through extraction pipelines. Although this dataset is relatively small comprising of 432 documents in total, however to the best of our knowledge this is the first of its kind which contains significant amount of conflicting information and has annotation for both event and relation arguments as well as the observation uncertainties. In addition to mention annotations, LDC provided a KG with Knowledge Elements (KEs) representing entities, events and relations. The LDC mentions were grounded to these KEs. We used the LDC grounding as the "gold" standard to evaluate our system’s grounding accuracy.

<table>
<thead>
<tr>
<th>Topic</th>
<th>English</th>
<th>Russian</th>
<th>Ukrainian</th>
</tr>
</thead>
<tbody>
<tr>
<td>T101</td>
<td>51</td>
<td>35</td>
<td>64</td>
</tr>
<tr>
<td>T102</td>
<td>56</td>
<td>51</td>
<td>35</td>
</tr>
<tr>
<td>T103</td>
<td>57</td>
<td>64</td>
<td>19</td>
</tr>
</tbody>
</table>

Table 1: Number of documents per topic per language

4.2 Experimental setup

We construct the full knowledge graph in two stages, as per the SM-KBP track requirements. First, a document-specific KG is constructed using the annotated mentions. Next, using the document-specific KG and cross-lingual entity co-reference as inputs, the final KG is constructed. The system overview can be seen in Figure 2. As described in Section 2.3, the evidence graph constructed using the inputs is grounded to the KG via random variables. For document-specific KG construction, where the inputs are mentions, the priors for their evidence random variables are assigned based on the observation uncertainties — 0.6 for event/argument roles annotated as hedged, 0.2 for negation and 1.0 otherwise. At the event/relation level, the hedging and negation information is encoded as priors of the target variable \(T\) in Figure 4. In an ideal case, where observation \(O1\) is not hedged or negated, \(T\) has a uniform prior implying that all existing KEs (\(Truck1, Truck2\)) and the new possibility \(Truck3\) are equally likely \(\{T = Truck1 : 0.33, Truck2 : 0.33, Truck3 : 0.33\}\). However, if \(O1\) is hedged, the prior for the new possibility \(Truck3\) is lowered, \(\{Truck1 : 0.41, Truck2 : 0.41, Truck3 : 0.18\}\). The intu-

---

4. Linguistic Data Consortium
ition being that if the new observation is uncertain then it shouldn’t be trusted as much as the existing KEs. For the latter scenario of full KB construction, the posteriors of the document-specific KGS are used as priors itself.

We consider the following two experimental settings for full KG construction:

1. Modeling only mention uncertainty i.e assuming all information sources are reliable. This is the setting of the SM-KBP 2018 workshop. Since there has been no published work on this dataset yet, we currently report only our results. Our LEAPFROG system has been submitted to the workshop and we are currently awaiting the evaluation results on the test data.

2. Modeling both source reliability and mention uncertainty by assigning varying degrees of trustworthiness score to the document sources.

4.3 Results

The results for the document-specific KG can be seen in Table 2. The metrics for evaluation are precision and recall, which check the grounding for each mention for entity, event and relations.

\[
P = \frac{\text{Exact Mention Match}}{\text{Predicted KEs}} \quad \quad R = \frac{\text{Exact Mention Match}}{\text{Gold KEs}}
\]

where Exact Mention Match is an indicator function which returns 1 if all the mentions grounded to a particular KE match exactly the gold KE’s mentions. Given that we use gold mentions as inputs, one would expect 100% precision. However, the gold knowledge elements, as provided by LDC, contain NIL clusters which are elements that do not have any grounding in the gold KG. Our model doesn’t handle NIL clusters for two main reasons: a) the model is trained in an unsupervised fashion and hence doesn’t have access to any training data for classifying NIL clusters, and b) since we are dealing with conflicting information we do not want to discard any elements early on. In Table 2, we report results ignoring the NILs and we see that as expected for entities the precision is close to 100%. For events and relations, we observe a similar improvement. We note that we use gold mentions but we do not use the gold KEs, so within and cross-document entity/event/relation co-reference is purely handled by LEAPFROG along with cross-lingual entity/event/relation co-reference.

<table>
<thead>
<tr>
<th></th>
<th>Entity</th>
<th>Event</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc-sp KG</td>
<td>79.78 / 98.58 / 88.18</td>
<td>52.91 / 79.50 / 63.53</td>
<td>41.89 / 83.22 / 55.72</td>
</tr>
<tr>
<td>doc-sp KG w/o NIL</td>
<td>98.51 / 98.47 / <strong>98.48</strong></td>
<td>77.11 / 80.21 / <strong>78.62</strong></td>
<td>77.02 / 85.07 / <strong>80.84</strong></td>
</tr>
</tbody>
</table>

Table 2: Precision / Recall / F1 for document-specific (doc-sp) KG with and without NIL clusters

Table 4 shows the results for the full KG in SM-KBP setting (Unbiased-sources). Unbiased-sources setting means all document sources are considered equally reliable. We report the
results ignoring the NIL clusters, for the same reasons discussed above. Since we construct the factor graph over a relevant candidate set, usually consisting of less than 10 candidates, LEAPFROG runs within minutes (~600sec) for the full KG construction.

For the second experimental setting, we assign reliability or trustworthiness scores for each document source. Document sources are the websites/news agencies/media from which the documents were extracted. Since the topics are all centered around Russian-Ukrainian conflict, one simple way to segregate the sources is country wise: Russian-based, Ukrainian-based and rest of the world-based. For example, Russian-based sources include websites/news agencies/media based in Russia like Russia Today (RT), Freedom Russia, etc. Details on the number of documents per source-type per language can be seen in Table 3. For each source-type setting, only the documents extracted from that set of sources are considered 100% reliable, and the rest documents are considered only 10% reliable. For example, in the Russian-sources setting, only documents extracted from the Russian-based sources are considered 100% reliable. All the four settings use the same set of documents for KG construction, just the reliability of documents vary across the different settings. The precision and recall number for all the settings can be found in Table 4. The results of the country-specific source settings are comparable to the unbiased setting. Biasing based on source reliability shifts the probability distribution amongst the different alternatives and as expected it doesn’t affect the grounding. In the next section, we study the interesting implications of these country-specific settings.

<table>
<thead>
<tr>
<th>Sources \ Lang.</th>
<th>English</th>
<th>Russian</th>
<th>Ukrainian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russian</td>
<td>12</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Ukrainian</td>
<td>18</td>
<td>38</td>
<td>50</td>
</tr>
<tr>
<td>RestWorld</td>
<td>100</td>
<td>134</td>
<td>58</td>
</tr>
</tbody>
</table>

Table 3: Number of documents per the three source settings across languages.

### 4.4 Case study: shooting of MH17

We discuss the effects of modeling mention and source reliability for one of the major disputed event, shooting of flight MH17, due to brevity of space.

<table>
<thead>
<tr>
<th>Sources</th>
<th>Entity</th>
<th>Event</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unbiased</td>
<td>36.63 / 56.29 / 44.38</td>
<td>29.62 / 56.93 / 38.96</td>
<td>48.96 / 49.65 / 48.80</td>
</tr>
<tr>
<td>Russian</td>
<td>36.74 / 56.46 / 44.51</td>
<td>29.75 / 56.22 / 38.90</td>
<td>48.96 / 49.65 / 48.80</td>
</tr>
<tr>
<td>Ukrainian</td>
<td>36.66 / 56.46 / 44.45</td>
<td>29.66 / 56.58 / 38.91</td>
<td>48.96 / 49.65 / 48.80</td>
</tr>
<tr>
<td>RestWorld</td>
<td>36.63 / 56.29 / 44.38</td>
<td>29.71 / 56.58 / 38.96</td>
<td>48.96 / 49.65 / 48.80</td>
</tr>
</tbody>
</table>

Table 4: Precision / Recall / F1 for full KG across different source settings. Unbiased-sources is the SM-KBP setting where all sources are equally reliable.
### Figure 6: Comparison of information interpretations across different source settings.

#### 4.4.1 Uncertainty analysis

Figure 6 shows some of the salient alternative interpretations generated by our model based on the different country-specific settings. And since BP performs entity, event and relation co-reference as part of the conflict resolution process, we observe how events and relation across languages nicely conflate. For the purpose of easy readability, we only show a few most likely alternative hypotheses.

In the unbiased-sources setting, where all the sources are considered equally trustworthy, we observe that the most likely agent of the attack is \{Russian air defense\}, with a very low probability for \{Ukrainian rebels\}, a second alternative. The remaining probability mass of 0.004 is for the *other, meaning there is a 0.004 probability of agent being something other than the possibilities discussed in the provided documents. However, when we trust only the Ukrainian-based sources, we see that the *other category gets a high probability mass, possibly due to two reasons:

1. There was not enough evidence supporting any one interpretation.
2. There were interpretations which nullified each other.

On manual inspection of the relevant events in Ukrainian sources, it was discovered that there were 11 potential agents mentioned by these sources, of which 10 agents didn’t have strong supporting evidence across the different documents. Only \{Separatists\} was one interpretation which was supported by two of the documents. However, the effect didn’t
propagate because there was also the observation \(\text{Separatists were not responsible}\), which shifted the probability mass to *other*. As we discussed earlier in Section 3.1, when an observation uncertainty increases (like negating), the mass tends to concentrate towards the *other* possibility. On the other hand, the RestWorld-sources consider \{Ukrainian rebels\} as the most likely hypotheses for the agent, \{Russia\} with 0.160 probability and an *other* possibility with 0.191. Since the dataset is relatively small, the probability distributions are sensitive to individual documents.

With respect to the instrument of the attack, we observe that all the four perspectives are consistent. However, another point of dispute is the home base location of this instrument. As seen in Figure 6, the unbiased, Ukrainian and RestWorld sources predominantly support the hypothesis of *Kursk*, a Russian town, being the base of the missile. Whereas the Russian sources strongly believe the missile to be positioned in *Pervomaisky*, a Ukrainian city.

This study of biasing using source reliability shows how important it is to consider the veracity of the information before propagating it downstream. Different sources have varying perspectives and how much we trust them strongly affects our belief.

4.4.2 Error Analysis

Based on our qualitative analysis, we bucket the possible grounding errors:

- **Common-sense and world knowledge**: Currently, the model uses only the provided information for constructing the KG. However, there are entities which have multiple identifiers that can’t be captured without external knowledge. For instance, the flight MH17 is referred as \{MH17, Malaysian passenger plane, Malaysian flight 17, Flight AOJ92C\}. Without prior knowledge, it is difficult to conflate these entities, further affecting the event and relation resolution.

- **Merging knowledge elements**: Currently, LEAPFROG merges existing KEs based on external knowledge in the form of cross-lingual entity co-reference, but doesn’t leverage the knowledge it has acquired so far. So, KEs like *BUK missile* and *SA-11 Buk missile systems* which refer to the same missile are not merged in the current setting, even though they always seem to occur under similar circumstances. Some higher-order inference is required on top the current LEAPFROG’s implicit co-referencing to identify such merges.

5. Related work

**Knowledge base construction**: Several approaches have explored KB construction, ranging from rule-based systems [Krishnamurthy et al., 2009] to using machine learning [Mitchell et al., 2018];[Etzioni et al., 2004]. [Poon and Domingos, 2007] use a Markov Logic Network [Domingos and Lowd, 2009] for information extraction. [Pujara et al., 2013, Pujara, 2016] explore probabilistic models to construct KG over noisy entity and relation extraction, specifically using probabilistic soft logic. Our work differs in that we use factor graph, a more generic framework which allows modeling different kinds of uncertainties apart from noisy extraction, to capture dependencies between variables. Furthermore, our approach employs belief propagation for entity, event and relation resolution and co-reference across multiple documents as well as languages. DeepDive [Shin et al., 2015] combines database
and statistical machine learning and is perhaps the most similar to our work in terms of using factor graphs for capturing relationships between entities. DeepDive resolves conflicting relations using a simple voting strategy which doesn’t account for the information source. This suggests that if a mention, cited by the same source, was present 100 times across the documents. Using a majority vote strategy, that interpretation would get a higher belief. This is misleading as those duplicate mentions are not adding any new information. By considering source information in our proposed method, we circumvent this problem.

**Probabilistic inference:** Existing systems construct a factor graph over a probabilistic KB to capture the correlations between entities. ARCHIMEDES [Chen et al., 2017] use the factor graph for establishing relations among facts and expands the KB by inferring more relations using Horn clauses [Chen and Wang, 2014]. Various inference algorithms have been employed for computing the posteriors. DeepDive experiments with approximate inference algorithms like Gibbs sampling [Pearl, 2014] and variational methods [Wainwright et al., 2008]. ARCHIMEDES uses MCMC [Niu et al., 2011]. We use belief propagation since it gives an exact inference, when there are no loops in the graph.

Our work primarily differs from the prior work in that we introduce a system that allows for modeling all - entities, events and relations in the same framework across languages. Limited work has been done for constructing KGs over events. Wu et al. [2015] build storylines for events from news articles, however they don’t capture alternative interpretations in those. Kuzey and Weikum [2014] mostly focus on extracting named entities from the news articles to populate the KB. To the best of our knowledge, we are the first to model events in a probabilistic KG.

### 6. Conclusion and Future Work

We introduce LEAPFROG, a probabilistic framework for automated knowledge graph construction over events and relations across languages. The ability of LEAPFROG to model different kinds of uncertainties in the data and maintain alternative interpretations of conflicting events/relations is a novel contribution. Experimental results show interesting implications of modeling source reliability and how the different interpretations of the same information vary across sources. Future work will be focused towards incorporating common-sense and world knowledge into LEAPFROG for a more informed inference. We are also extending the current LEAPFROG probabilistic inference engine to include variational particles to deal with loopy belief graphs. We hope that LEAPFROG spurs interesting discussions on using probabilistic models for inference over conflicting information, in the community.

**References**


