Abstract

Information extraction from text, IE for short, is the backbone of automated knowledge base construction. IE usually comes with precision estimates; however, it lacks awareness of recall. This paper introduces and discusses the issue of recall estimation and its practical importance. We present RECALLIE, a methodology for estimating the possible recall from a given text segment. RECALLIE uses distant supervision to estimate from language features whether a passage contains all objects for a given subject-predicate pair. We evaluate RECALLIE across various granularities of text, and across various predicates. Our preliminary results indicate that estimating recall is a promising direction and technically feasible.

1. Introduction

Knowledge bases containing general world knowledge or domain knowledge are major building blocks for semantic search, question answering, intelligent assistants, and other AI applications. The construction of knowledge bases from textual sources, also known as textual information extraction (IE), is a long-standing goal with many efforts in the natural language processing, machine learning and data management communities. Although IE methods have made substantial progress over nearly two decades, their outputs and the resulting knowledge bases are nevertheless decidedly incomplete, typically containing only a tiny subset of the facts that a human would consider relevant.

Textual IE usually produces (ideally canonicalized) subject-predicate-object triples such as ⟨Trump, hasChild, Eric⟩, and annotates them with estimates of precision (also called accuracy, correctness or confidence), e.g., a 93% belief that Eric is really Trump’s child. In contrast, IE usually lacks such an ability for recall (also called completeness or coverage). It is not able to quantify whether it might have extracted all facts pertaining to a certain topic, e.g., all children of Donald Trump (see Fig. 1).

Importance of recall information  Recall-awareness of IE is a crucial and highly desirable property for a variety of downstream use cases, including question answering, curation prioritization and automated knowledge base construction.
Today’s question answering systems are well geared for questions where exactly one answer should be returned (e.g., quiz questions or reading-comprehension tasks) (Fader et al., 2014; Yang et al., 2015). In contrast, for questions with sets of answers, QA systems often merely yield subsets; for example, Google returns only 43 results for “Universities and colleges in Massachusetts”, while Wikipedia lists 110. In benchmarks such partial answers are often considered good enough, but this is far from the desired behavior in real applications. State-of-the-art QA also struggles with questions that have no answer, usually still returning a best-effort answer even if it is incorrect (e.g., children of Angela Merkel – she has none). Recall information would enable a better treatment of such questions.

Guiding editors in how to prioritize curation efforts is a key issue for collaboratively built and maintained knowledge bases such as Wikidata (Balaraman et al., 2018; Darari et al., 2017). Yet, methods to automatically identify incomplete parts are still largely based on aggregate-level statistics, and not specific to individual entities (Razniewski et al., 2017; Galárraga et al., 2017). Recall information from text could both increase the specificity of such methods, and enhance them with text-based explanations.

Quantifying recall would also be useful to automated knowledge base construction techniques in two regards: i) to dynamically adjust confidence thresholds, i.e., lowering thresholds in case of missing information and increasing thresholds in case of too many extractions, and ii) to reallocate search budgets to incomplete regions, while stopping the exploration of complete areas (Ipeirotis et al., 2007; Jain et al., 2008).

Contribution and Approach In this paper we present RECALLIE, a methodology for estimating the recall of textual information extraction. RECALLIE uses distant supervision to estimate from textual features whether a text segment contains all objects for a given subject-predicate pair, e.g., whether a given text mentions all children of Trump (cf. Figure 1). For an experimental study, we seed RECALLIE with fact counts for 5 Wikidata relations as ground truth. Using these counts, we train and evaluate on Wikipedia-extracted and Web-extracted sentences and paragraphs, finding that recall estimation is generally feasible and yields informative assessments.

Our conceptual contributions are:

- We introduce and define the novel problem of textual recall estimation, and we discuss its key features.
We present a methodology, **RECALLIE**, along with experimental results that demonstrate its practical value.

Our experiments with **RECALLIE** lead to the following technical insights:

- **Predicates**: Recall estimation is feasible for a diverse set of predicates, ranging from family relations to organizational membership and group compositions.
- **Text units**: Recall estimation is feasible both on the level of sentences and paragraphs.
- **Text sources and classifiers**: Texts from Wikipedia articles and from Web pages can both be handled with satisfying performance. Learning word embeddings for specific corpora leads to significantly better results than using pre-trained GloVe or word2vec models.

2. **Background and Use Cases**

2.1 **Automatic Knowledge Base Construction**

Automated knowledge base construction (AKBC) using textual information extraction has become a mainstream technique in recent years, being used by major tech companies such as Google, Amazon or Alibaba, by news agencies such as Bloomberg, and being offered to other industries via specialized companies such as Diffbot, Metaphacts or Lattice (now part of Apple).

AKBC is usually based on explicit or latent patterns, and the duality of patterns and facts (Brin, 1998) enables to iterate pattern discovery and application processes. Most AKBC frameworks annotate their extraction results with confidence or precision scores. The NELL framework (Carlson et al., 2010), for instance, has 75% confidence in the fact that Trump’s spouse is Melania Knauss.\(^1\) Similarly, the DeepDive extraction and reasoning framework annotates each extracted fact with a marginal probability that represents the systems confidence in the fact (Shin et al., 2015). Such confidence scores are illustrated in Figure 1.

None of the state-of-the-art systems yields recall estimates, i.e., gives information on whether the system believes to have extracted all facts on a certain topic. Recall of IE has so far been only evaluated post-hoc, i.e., using benchmark datasets where it is known which facts should be extracted (for a listing of some post-hoc evaluations, see Niklaus et al., 2018).

2.2 **Related Work on Information Extraction**

Information extraction from text sources has been greatly advanced over the past two decades; see, e.g., (Agichtein and Gravano, 2000; Etzioni et al., 2004; Suchanek et al., 2009; Mintz et al., 2009; Riedel et al., 2013; Dong et al., 2014; Shin et al., 2015; Mausam, 2016; Chiticariu et al., 2018; Stanovsky et al., 2018). The underlying methodologies span regular-expression matching, rule-based extraction, conditional random fields, constraint reasoning, all the way to deep learning. Depending on the task at hand, IE often achieves high precision (sometimes above 90%). However, evaluating its recall is inherently hard, as this would

---

\(^1\) [http://rtw.ml.cmu.edu/rtw/kbbrowser/personeurope:donald_trump](http://rtw.ml.cmu.edu/rtw/kbbrowser/personeurope:donald_trump)
require exhaustively annotated corpora as gold standard. As a consequence, assessing and optimizing recall has typically been an afterthought at best, and is usually completely disregarded.

In contrast, recall is one of the key metrics in information retrieval (IR), i.e., in search applications. Here, recall is measured in terms of retrieving a large fraction of the relevant documents or passages, where relevance is stated by gold-standard annotations. In the context of entire IE workflows (e.g., for text analytics over business news), the prior work of (Ipeirotis et al., 2007; Jain et al., 2008) has considered optimizations for recall. However, this solely refers to the search-centric parts of such workflows, that is, the document or passage sets that are then fed into IE steps.

Linguistic theories, such as Grice’s maxims of cooperative communication (Grice, 1975) suggest that communication can give hints towards recall. Consider the two sentences in Fig. 2 (left side). Both sentences allow to extract two children facts. Yet, for the first sentence, it is much more plausible that Tom and Susan are all children of John, than for Lisa and Bob w.r.t. Mary in the second sentence. Based on the context provided there, it could well be the case that Mary has other children that are merely too young or too old to be brought to school.

The closest to recall-aware information extraction is recent work on counting quantifier extraction (Mirza et al., 2018). Relation counts are extracted from phrases such as “Jolie has six children”, which, in a second step, are compared against fact counts in knowledge bases. In contrast, our work presented in this paper aims to enhance the fact extraction process itself with recall awareness.

2.3 Use Cases for Recall-aware IE

Recall assessment is straightforward for properties that are both mandatory and single-valued, e.g., dateOfBirth: recall is 1 exactly if a value can be extracted, 0 otherwise. For all other properties, i.e., those that are not mandatory (e.g., dateOfDeath), or those that can take multiple values (e.g., child), recall is of interest in various use cases.

Recall for question answering Question answering has made impressive progress in recent years, with IBM’s Watson (IBM, 2012) beating humans in the popular Jeopardy game, personal assistants such as Siri and Alexa becoming more and more widespread, and
search engines incrementally increasing the QA component in favor of traditional document retrieval.

However, these advances are often focused on easy-to-answer popular themes, or in the case of Watson, still benefit from the design paradigm that there is exactly one correct answer per question. Real-world questions, however, frequently have many answers, and it is often important to know about the coverage of the result, to plan further data acquisition. For instance, if a question answering system is aware that it knows only 43 out of 110 universities and colleges in Massachusetts, but 24 out of 24 in South Dakota, a user may accordingly consult other resources in the first case, while continue working with the answer in the second case.

This challenge becomes particularly pronounced in the case of question that have empty answers, e.g., “Which Antibiotics can treat Hepatitis?” (a viral infection). Due to the incompleteness of knowledge bases, manifested in the open-world assumption, today’s KBs cannot confidently argue that certain facts do not hold, and therefore often return highest ranking answers, where no answer is correct. This also holds for claim verification tools, that typically focus on language cues and context. Given a simple incorrect claims such as Trump is the president of China, the CredEye tool (Popat et al., 2018) gives a 56% confidence for correctness. Recall information, in this case, knowing that all public offices held by Trump are known, would help to assess the validity of such statements with near-perfect confidence.

Recall for curation prioritization For collaboratively created knowledge bases such as Wikidata (Vrandečić and Krötzsch, 2014), scalability and efficient use of human resources are major issues. Knowing where to focus efforts is complicated because of the continuous, incremental data insertions by its users, the distributed expertise and interest of the community, and the absence of a defined boundary in terms of the scope of the KB. There exists one tool, COOL-WD (Darari et al., 2017), which empower editors to manually assert completeness for subject-property-pairs, and another tool, Recoin (Balaraman et al., 2018), that uses frequency statistics in order to rank missing properties on entity pages. Text-extracted recall information could here be a great help to derive more specific suggestions on where data might be complete, or incomplete.

Recall for information extraction Information extraction typically operates under resources limitations: Only fragments of available datasets can be analyzed (e.g., even Google would until recently only index the first 10 MB of a webpage\(^2\)). Resource limitations become even more an issue when freshness matters, as in the case of trending topics, or in the case of on-the-fly KB construction (Nguyen et al., 2017). Information extraction thus necessarily has to make choices on how to best allocate its resources. While there exists some suggestions for text-database query optimization based on sampling (Ipeirotis et al., 2007; Jain et al., 2008), or species-sampling-based recall estimation techniques (Trushkowsky et al., 2013; Salloum et al., 2013), recall knowledge gained at extraction time could much more directly steer the extraction process, by early stopping exploration in case of sufficient recall, or in turn by increasing the exploration depth if recall is low.

\[^2\] https://productforums.google.com/forum/#!topic/webmasters/BsLw3Yd-RKg
3. Problem and Approach

While information extraction is a noisy process with both false positives and false negatives, our focus here is on whether, in principle, a text segment allows the extraction of all facts that hold in reality. For this purpose, we assume we have perfect knowledge of all real-world facts for the objects that are connected to a specific subject \( s \) and property \( p \); we denote this object set as \( RW\{o \mid sp\} \). Now assume an educated and linguistically versed human is presented with a text segment \( t \) and the task of telling which objects \( o \) she would assign to a fixed subject \( s \) and property \( p \) given solely the text \( t \) (i.e., without knowing the real-world facts). We denote this ground-truth object set as \( GT\{o \mid sp,t\} \).

**Text Recall Assessment Problem.** Given subject \( s \), property \( p \), and text segment \( t \), predict whether the human ground-truth from reading \( t \) covers all real-world facts:

\[
GT\{o \mid sp,t\} \supseteq RW\{o \mid sp\}.
\]

Note that this problem is different from assessing the quality of specific IE methods and tools. Since there is no perfect IE, considering the extractions from an IE tool would confound two distinct issues: 1) whether a text segment contains all information of interest (our present problem), and 2) what the recall of the specific IE tool is (a standard evaluation criterion for IE methods). Although our automated evaluation (see Section 5) necessarily builds on concrete choices for IE methods, we keep this separate from the fundamental problem of recall assessment given solely a text segment, as described above.

By casting the problem into a binary classification task, we look only at two cases: a) \( GT\{o \mid sp,t\} \) contains all real-world facts, and b) it does not. This formulation disregards complex graded cases, such as \( GT\{o \mid sp,t\} \) containing at least 70% of the real-world facts. Nevertheless, the problem naturally invites the use of scores that are confidences/probabilities. For example, for the first sentence in Figure 2, the probability to contain all \( \langle \text{John, child, *} \rangle \)-facts might be 0.9, while for the second sentence, the probability to contain all \( \langle \text{Mary, child, *} \rangle \) facts might be 0.4. An illustration of how such confidence scores could be applied to Web search snippets is shown in Figure 3.

**IE method**  IE recall assessment may be highly dependent on the actual IE method used to extract facts. In this work we approximate the ideal output of an IE method via the combination of open information extraction (Open IE) and object label matching. Note that this specific choice is not decisive for our approach and merely serves as a concrete instantiation of our framework. Our focus is on the recall estimation, and other IE methods could be plugged in as well.

To evaluate whether a text snippet contains an \( \langle s, p, o \rangle \)-fact, we identify the alias names \( AN \) of the true objects \( O \) from an existing knowledge base. To extract facts from text snippets, we rely on the open information extraction system OpenIE 4 (Pal and Mausam, 2016). As text snippets, we consider a text dedicated to a considered subject \( s \), for instance, the Wikipedia page of the subject or text results from web search given the subject as a query. For each triple extracted by open information extraction, we then check whether (i) the arguments of the triple contain one of the possible object aliases in \( AN \), and (ii) they contain any of the predefined terms related to the predicate of interest. The predefined list of terms related to each predicate contains terms that are manually selected from the
RecallIE: Making information extraction recall-aware

Figure 3: Possible application of a recall classifier on web search snippets.

PATTY paraphrase dictionary (Nakashole et al., 2012). If both conditions are satisfied, we consider the fact as the ideal IE-extracted fact, and the text snippet as the text provenance.

For example, suppose we are interested in extracting facts for the hasChild property for the subject Angelina Jolie. OpenIE extracts the triple \( \langle \text{Maddy}, \text{is first adopted son [of]}, \text{Jolie} \rangle \) from the text snippet “Jolie’s first adopted son is Maddy.” As (i) Maddy is one of the aliases of Angelina Jolie’s child Maddox Chivan in IMDb, and (ii) son appears in the predefined list of terms for the child predicate, we consider that the text snippet contains the fact \( \langle \text{Angelina Jolie}, \text{hasChild}, \text{Maddox Chivan} \rangle \).

**Features** Textual recall assessment can be approached via two main classes of features: (i) textual features and (ii) contextual features. Textual features refer to any content appearing in the piece of text under consideration, while context refers to information in surrounding segments, background knowledge about entities, and metadata such as source reliability or timestamps. As textual features are universal, we consider only textual features in this paper, although contextual features are a natural and interesting extension. To describe textual features, we use either n-grams or word embeddings.

**Pipeline** The full pipeline of RecallIE is shown in Figure 4. In the first step, for given subject entities, relevant text fragments are extracted from Wikipedia or the Web. In the second step, the fragments are transformed into triple sets by using OpenIE. In the third step, object aliases and predicate paraphrases are matched against the triples, and matches are used to label segments in the fourth step as complete or incomplete. In the fifth step, segments are masked. Subsequently, data is split into train/test sets, and classification is run and evaluated.
4. Experimental Setup

**Predicates** We perform experiments for 5 Wikidata predicates that span three different domains:

2. Education and work: *educatedAt* (P69), *employer* (P108).
3. Band compositions: *has part* (P527) for instances of the *musical ensemble* (Q2088357) class, henceforth called *member*.

**Labelled Data** We use distant supervision to automatically label data. Assuming that Wikidata’s fact coverage is near-perfect for popular entities, and hence simulating full real-world coverage, for each of the 5 predicates above, we collect the 1000-8000 most popular subjects in Wikidata, along with their facts for the respective property. Subject popularity is determined by the number of facts stored in Wikidata. Table 1 (top rows) lists the number of subjects each relation has in Wikidata, along with the number of subjects that have at least 2 objects, and the number of subjects used for generating labelled data.

For every subject, we then collect text snippets about the subject from which facts are extracted according to the IE method explained in Section 3. A text snippet is considered to be complete, and labelled as such, if we can match alias names for all objects found in Wikidata. It is labelled as incomplete otherwise. We publish our labelled data at [anonymized URL].

**Text preprocessing** To avoid having classifiers that put more importance to features like the appearance of a certain proper name or a certain number in the text (e.g., ‘John’ or ‘two’) over completeness cues such as ‘only’ or ‘all’, we mask proper names and numbers, indicated by *NNP* and *CD* part-of-speech tags respectively, with generic placeholders. For instance, “Jolie’s first son is Maddox.” would become “⟨propername⟩’s first son is ⟨propername⟩.” We also remove stopwords, except from the input text for LSTMs.
RecallIE: Making information extraction recall-aware

Table 1: Number of Wikidata facts and derived labelled text segments. +/- signifies complete/incomplete w.r.t. facts found.

<table>
<thead>
<tr>
<th></th>
<th>child</th>
<th>spouse</th>
<th>member</th>
<th>employer</th>
<th>educatedAt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wikidata #subj</td>
<td>40,145</td>
<td>45,261</td>
<td>8,901</td>
<td>58,731</td>
<td>273,128</td>
</tr>
<tr>
<td>Wikidata #subj w/ ≥2 obj</td>
<td>15,022</td>
<td>4,055</td>
<td>1,022</td>
<td>12,885</td>
<td>72,847</td>
</tr>
<tr>
<td>Wikidata seeds</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>8,000</td>
</tr>
<tr>
<td>Wikipedia Sentences +/-</td>
<td>135/ 2,050</td>
<td>119/ 2,444</td>
<td>672/10,358</td>
<td>47/1,499</td>
<td>447/ 2,603</td>
</tr>
<tr>
<td>Paragraphs +/-</td>
<td>217/ 1,595</td>
<td>385/ 2,044</td>
<td>930/ 5,362</td>
<td>108/1,248</td>
<td>339/ 2,384</td>
</tr>
<tr>
<td>Web Sentences +/-</td>
<td>5,609/11,089</td>
<td>673/11,068</td>
<td>290/ 6,687</td>
<td>53/2,240</td>
<td>185/ 1,459</td>
</tr>
<tr>
<td>Paragraphs +/-</td>
<td>1,184/ 2,774</td>
<td>151/ 3,369</td>
<td>176/ 4,932</td>
<td>30/2,052</td>
<td>159/ 1,367</td>
</tr>
</tbody>
</table>

Baselines We employ two baselines. The first baseline, length, classifies the longest text segments, measured by character length, as complete. The second baseline, #pnames, counts the number of proper names per segment, and classifies those containing the most as complete. For both baselines, the classification threshold is chosen so that the ground truth class distribution is maintained, therefore precision and recall coincide.

Classification methods To build the classifiers, we leverage the state-of-the-art methods for text classification, namely feature-based Support Vector Machines (SVMs) as well as neural-based Long Short-Term Memory networks (LSTMs) (Tai et al., 2015), which gain popularity in recent years for automatic text processing.

- We employ SVMs with linear kernel using both unigrams and bigrams as features, weighted simply by term frequency.
- LSTMs are used to encode sentences/paragraphs, using word representations (dimension $d = 100$) that are learned from scratch (initialized uniformly). One hidden layer of size 256 with ReLU activation and an output layer with sigmoid activation are used for binary classification. We employ the Adam optimizer with default parameter values. The models were trained for 20 epochs.

Text granularity and source We investigate two granularities of text units: sentences and paragraphs. We also investigate two kinds of sources for texts: (i) as high-quality but smaller source the Wikipedia article of the respective subjects, (ii) as noisier but larger text collection general web search result. The results were obtained using the Bing Search API, querying with subject names and relevant predicate paraphrases. Per subject, 5 to 6 queries were run, and the first 10 results for each query were retained. Text from the respective websites was extracted, after removing duplicate results, and Wikipedia articles.

The total size of the labelled text corpora per predicate is shown in Table 1 (bottom rows). In each cell, the first number indicates the number of complete (+) text units, while the second number indicates the number of incomplete (-) ones. We report only sentences and paragraphs that contain at least one fact (those containing no facts at all are much more numerous, but not helpful in training). As one can see, the training corpora are highly imbalanced, with incomplete instances outnumbering complete ones by a factor of 2 to 70.
5. Evaluation

We conduct three kinds of evaluations: An automated evaluation using the automatically generated labelled data, a manual evaluation based on human judgment, and an extrinsic evaluation in an information extraction use case. In addition, we inspect relevant features for classification decisions.

5.1 Automated Evaluation

We randomly split the labelled data into training (80%) and testing (20%) sets, retaining the class distribution. For this experiment, we use labelled text segments from Wikipedia as our dataset. Table 2 shows the performance comparison of feature-based SVM and neural-based LSTM classifiers evaluated across properties on different text units. We report precision (P), recall (R) and F1-score (F) of the models in terms of identifying complete text segments, i.e., computed over only complete samples in the testing set.

Both SVMs and LSTMs outperform the two baselines based on text length and #pnames found in the text, by a considerable margin. For the sentence level, LSTMs generally outperform SVMs possibly due to their capability of encoding sequence of words. For the paragraph level, LSTMs are able to identify more complete paragraphs, shown by their higher recall. SVMs on the other hand yield better precision, which results in better overall performance (F1-score) particularly for spouse and member predicates.

In general, both models perform best on identifying complete paragraphs for the spouse property with .73 and .70 F1-scores by SVM and LSTM, respectively, and SVM for the educated at property with .63 F1-score. Meanwhile, both models fail at identifying complete sentences for the employer predicate, presumably because it is a very rare case that all employments are listed in the same sentence. For instance, it is more common to find “He served as a professor at [University A], and also held an appointment at [University B]. In July 2007, he left [University A] and joined the faculty of [University C]. He was also a visiting professor at the [University D],” as a complete paragraph. Even though the member predicate has the most sentences/paragraphs labelled as complete, performance of both SVMs and LSTMs are mediocre, showing the difficulty of the task for this particular predicate possibly due to noisy training data.

Table 2: Performance of SVMs and LSTMs across different properties.

<table>
<thead>
<tr>
<th>Text unit</th>
<th>Model</th>
<th>child</th>
<th>spouse</th>
<th>member</th>
<th>employer</th>
<th>educatedAt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Sentence</td>
<td>length</td>
<td>.05</td>
<td>.28</td>
<td>.13</td>
<td>0</td>
<td>.24</td>
</tr>
<tr>
<td></td>
<td>#pnames</td>
<td>.05</td>
<td>.22</td>
<td>.17</td>
<td>0</td>
<td>.28</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>.50</td>
<td>.46</td>
<td>.35</td>
<td>.33</td>
<td>.34</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>.47</td>
<td>.44</td>
<td>.45</td>
<td>.43</td>
<td>.38</td>
</tr>
<tr>
<td>Paragraph</td>
<td>length</td>
<td>.17</td>
<td>.37</td>
<td>.26</td>
<td>.10</td>
<td>.21</td>
</tr>
<tr>
<td></td>
<td>#pnames</td>
<td>.19</td>
<td>.40</td>
<td>.31</td>
<td>.20</td>
<td>.29</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>.50</td>
<td>.44</td>
<td>.47</td>
<td>.74</td>
<td>.73</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>.41</td>
<td>.83</td>
<td>.55</td>
<td>.54</td>
<td>1</td>
</tr>
</tbody>
</table>
RecallIE: Making information extraction recall-aware

<table>
<thead>
<tr>
<th>Text unit</th>
<th>Property</th>
<th>LSTM (from scratch)</th>
<th>LSTM (GloVe)</th>
<th>LSTM (word2vec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>Sentence</td>
<td>child</td>
<td>0.47</td>
<td>0.44</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>educated at</td>
<td>0.47</td>
<td>0.98</td>
<td>0.64</td>
</tr>
<tr>
<td>Paragraph</td>
<td>child</td>
<td>0.41</td>
<td>0.83</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>educated at</td>
<td>0.42</td>
<td>0.96</td>
<td>0.58</td>
</tr>
</tbody>
</table>

Table 3: Performance of LSTMs with different embeddings for child and educated at.

<table>
<thead>
<tr>
<th>Text unit</th>
<th>Text source</th>
<th>child</th>
<th>spouse</th>
<th>member</th>
<th>employer</th>
<th>educatedAt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Paragraph</td>
<td>Wikipedia</td>
<td>.41</td>
<td>.83</td>
<td>.55</td>
<td>.54</td>
<td>1.70</td>
</tr>
<tr>
<td></td>
<td>Web</td>
<td>.66</td>
<td>.49</td>
<td>.56</td>
<td>.12</td>
<td>.03</td>
</tr>
</tbody>
</table>

Table 4: Performance of LSTMs across properties using different textual sources.

We also investigated the impact of initializing the embedding layer of LSTMs with weights from pre-trained embeddings, namely (i) GloVe (Pennington et al., 2014) trained on Wikipedia + Gigaword (6B tokens) and (ii) word2vec (Mikolov et al., 2013) trained on our corpus. Results for two predicates, child and educatedAt, are presented in Table 3. Even though we observe some improvements, specifically by word2vec embeddings for the paragraph level, their contributions are not significant.

We next compare the performance of LSTMs on identifying complete paragraphs taken from different textual sources, i.e., Wikipedia vs Web search results. Results are reported in Table 4. In general, evaluation scores on the Web data are significantly worse than using Wikipedia as text source. Our intuition is that because texts from web are more noisy, learning patterns is more difficult. We have also attempted to counter the label bias in the training data by adjusting the sampling ratio of positive/negative cases. While this improves performance to some extent, it does not fundamentally overcome the label bias by the limited data availability.

5.2 Manual evaluation

To better understand how RecallIE made predictions, we had two annotators label the completeness of a balanced set of complete and incomplete sentences for two predicates, child and educatedAt, for 80 sentences in total. To simulate real usage of RecallIE, we included also the dominant class of sentences that contained only one object.

Annotators were allowed to assert arbitrary confidence scores between 0 and 1, and made extensive use of these, as from the limited context of sentences, it was seldomly possible to assert definite completeness or incompleteness. To pick up graded signals, in the following we report Pearson correlation coefficients.

We show some sample sentences along with the respective scores assigned by the annotators/RecallIE in Table 5. We found a good annotator agreement on child (0.83 correlation), and moderate agreement on educatedAt (0.57). Correlation between annotators
Anonymous authors

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Annotators score</th>
<th>RecallIE score</th>
</tr>
</thead>
<tbody>
<tr>
<td>He was the father of actor Pierre Renoir (1885-1952) filmmaker Jean</td>
<td>1</td>
<td>0.54</td>
</tr>
<tr>
<td>Renoir (1894-1979) and ceramic artist Claude Renoir (1901-1969).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>He is the father of actress Angelina Jolie and actor James Haven.</td>
<td>0.85</td>
<td>0.52</td>
</tr>
<tr>
<td>His only legitimate child Ada Lovelace is regarded as the first computer</td>
<td>0.25</td>
<td>0.49</td>
</tr>
<tr>
<td>programmer based on her notes for Charles Babbage’s Analytical Engine.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>His daughter Julie Gavras and his son Romain Gavras are also filmmakers.</td>
<td>0.75</td>
<td>0.46</td>
</tr>
<tr>
<td>Genghis Khan was aware of the friction between his sons (particularly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>between Chagatai and Jochi) and worried of possible conflict between</td>
<td></td>
<td></td>
</tr>
<tr>
<td>them if he died.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“From this moment I am no longer the king; the king is Victor my son.”</td>
<td>0.13</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Table 5: Example sentences along with human and RecallIE’s confidence in completeness.

<table>
<thead>
<tr>
<th>child</th>
<th>spouse</th>
<th>member</th>
<th>employer</th>
<th>educatedAt</th>
</tr>
</thead>
<tbody>
<tr>
<td>85 (propername)</td>
<td>50 married twice</td>
<td>65 (propername)</td>
<td>24 academic</td>
<td>74 educated (propername)</td>
</tr>
<tr>
<td>81 sons</td>
<td>44 children</td>
<td>64 (propername)</td>
<td>23 academic year</td>
<td>67 educated</td>
</tr>
<tr>
<td>42 (propername)</td>
<td>42 married</td>
<td>63 duo</td>
<td>21 faculty</td>
<td>56 briefly attended</td>
</tr>
<tr>
<td>41 (number)</td>
<td>39 list (number)</td>
<td>55 featuring lineup</td>
<td>20 (propername) academic</td>
<td>50 attended (propername)</td>
</tr>
<tr>
<td>40 (propername)</td>
<td>37 second marriage</td>
<td>54 (propername)</td>
<td>19 year (number)</td>
<td>47 briefly</td>
</tr>
<tr>
<td>33 (propername)</td>
<td>36 actor</td>
<td>38 (propername)</td>
<td>18 taught</td>
<td>33 left graduating</td>
</tr>
<tr>
<td>33 (number)</td>
<td>33 (propername)</td>
<td>33 consists</td>
<td>16 served (propername)</td>
<td>24 (propername) left</td>
</tr>
<tr>
<td>31 (propername)</td>
<td>33 musicians</td>
<td>33 consists (propername)</td>
<td>16 classical philology</td>
<td>24 office</td>
</tr>
<tr>
<td>28 (propername)</td>
<td>31 actress</td>
<td>32 (propername)</td>
<td>15 (propername) classical</td>
<td>23 (propername) raised</td>
</tr>
<tr>
<td>26 (number)</td>
<td>31 later married</td>
<td>30 vocals (propername)</td>
<td>15 studies (propername)</td>
<td>21 admitted</td>
</tr>
</tbody>
</table>

Table 6: Most important unigrams and bigrams for SVMs evaluated across properties.

and RecallIE was lower, at 0.40 for child and 0.16 for educatedAt. We suspect that the low agreement in part stems from biases in the data, for instance, it seems that (Wikidata) sports people and Renaissance persons would receive less formal education than contemporary politicians or scientists, so RecallIE may pick up cues towards professions and epochs. We investigate that next by taking a closer look at classification features.

5.3 Feature analysis

In Table 6 we show the most informative features for the n-gram-based SVMs on the paragraph level. As one can see, most unigrams become indicative because they are strongly related with the properties, e.g., sons, married, duo, educated and admitted, with a few exceptions that indeed indicate completeness like consists [of]3, which most of the time is followed by a complete list of names. Some unigrams reveal interesting bias from the data which becomes a proxy for indicating completeness, such as musicians and actress (for spouse) and academic and taught (for employer).

Most of the highly weighted bigrams signal the beginning of name listing, such as daughters (propername), lineup (propername) and educated [at] (propername). Some bigrams convey temporal information, such as later married, academic year, briefly attended and left graduating, which indicate that the paragraph contains a narrative that lists object names

3. Stop words are omitted as part of the preprocessing step.
for different time periods. This is particularly true for spouse, employer and educatedAt predicates, for which usually only one object is valid at a certain time.

5.4 Extrinsic use case

In this section we evaluate the usefulness of RecallIE for saving resources in automated knowledge base construction. We simulated knowledge base construction for 18/10 popular entities for the child/educatedAt predicates. For each of the subjects, we used 100 paragraphs as text segments, retrieved by web search via concatenating the entity name with “children”/“education”. We then used RecallIE’s recall scores for reranking the paragraphs, and investigated the number of facts found as a function of the number of processed paragraphs.

We compare with two baselines. The first baseline was to process articles simply in the order they were returned by Google, and paragraphs inside articles in consecutive order. As a second baseline we use topic modelling, where we ranked all paragraphs (across articles) by the fraction they were constituted by the topic that by manual inspection was found to be most relevant for each predicate. We used a Wikipedia-trained LDA model with 100 topics.

The results of the evaluation are shown in Figure 5. As one can see, inspecting paragraphs in the order of recall estimates leads to a substantial increase of facts discovered in the first few inspected paragraphs (e.g., 2.7 facts in the first 5 paragraphs) compared with inspecting them in the default order as supplied by the search engine and the page layouts. Nevertheless, ranking by recall does not outperform ranking by topic relatedness, which appears to be a strong baseline. More research, possibly on combining topical relatedness with recall, is needed here.

6. Discussion

In this section, we discuss several aspects around recall estimation, in particular, the hardness of the task and ongoing work on extensions.
**Task difficulty**  The prediction results, ranging in F1-score from 0 to .64 for the sentence level and .09 to .73 for the paragraph level, are significantly lower than typical scores in information extraction (e.g., up to .83 F1 in the cold-start knowledge base population TAC 2017 challenge (Getman et al., 2017)). Our experiments revealed that the following aspects contribute to the problem’s hardness:

- **Training data quality.** We find that distantly supervised training data for recall is much noisier than for classical IE tasks, because knowledge bases such as Wikidata, despite having low error rates, have many gaps where they are incomplete (rather than incorrect). For instance, if the knowledge base contains only two famous children of a subject, sentences like “She is very proud of her twins, John and Mary, who have successful acting careers.” is labelled as complete over sentences having more chance to be complete like “She gave birth to twins, John and Mary, and two sons, Bob and Dave.” This issue mirrors a similar problem found in (Mirza et al., 2018).

- **Low NED recall.** Our current pipeline requires to match text-extracted entities against knowledge base entities. Yet even famous subjects frequently have obscure objects, e.g., none of Bill Gates' children has a Wikipedia page. NED tools consequently often failed to correctly resolve such objects. In the present work we thus opted for lexical matches, trading a higher recall against a lower precision due to spurious name matches.

- **Imbalance and topical skew.** Wikidata is highly skewed towards the count “1”, e.g., 83%/74% of all Wikidata humans with education/child information have only one such fact. Consequently, many odd text segments are already complete, making the learning of useful patterns more difficult. Similarly, we suspect that relations are skewed towards certain professions or backgrounds, with sports people receiving less formal education, or with monarchs having especially many children. And while such skews add interesting facets about priors and external knowledge to the problem, on the text classification level, they negatively affect the quality of learning.

- **Time-variance.** While some KB relations are quite stable (e.g., children), others are more volatile, and may both grow or shrink over time (e.g., band membership) (Wijaya et al., 2015). Such dynamicity adds complexity to the recall assessment, as recall may then be specific to certain time points.

**Relative recall**  Our work so far has focused on estimating the recall w.r.t. reality, as judged gold-standard annotators (who, in principle, could have all kinds of encyclopedia, databases and the whole Web at their disposal). An equally important question, close to previous work on species-count estimation (Salloum et al., 2013), is to estimate the recall with regard to what could be maximally achieved by using the union of all possible sources. This is conceptually different insofar as a sentence “Besides his daughter Pythia, the names of Aristotle’s 7 children are unknown” does not contain complete information, but could be used to save further exploration efforts.

**Modelling and reasoning**  While we have shown that textual information can be useful in inferring the recall of extractions, recall estimation might benefit from more explicit modelling and reasoning.
One relevant aspect could be temporal reasoning. For instance, a professional career without temporal gaps (e.g., high school till 1993, BSc. 1994-1997, then launch of a startup in 1998) is a helpful indicator towards complete education extraction. Such reasoning could be applied on top of temporal information extraction (Ling and Weld, 2010).

Another aspect are statistical priors and typicality information. Information that rock bands often consist of one bassist, 1-2 guitarists, one vocalist and one drummer could be helpful in assessing extraction recall at extraction time, similar as done post-hoc in (Galárraga et al., 2017).

A third aspect could be numeric bounds for KB relations. If some text segment states “Trump has been married three times”, such numeric information could be used as explicit bound for extractions, i.e., terminating extractions after having found three facts, or labeling extractions that only find one fact as incomplete (Mirza et al., 2018).

**Other Datasets** So far, RECALLIE has been applied to Wikidata relations, which are manually designed. We would also like to apply it to latent/universal schemas (see, e.g., (Riedel et al., 2013)), or open information extraction (Fader et al., 2014). While absolute recall could be difficult to assert in such cases, relative recall w.r.t. other text sources would still be a very useful measure.

### 7. Conclusion

This paper presented a first endeavor on bringing recall into the foreground of research on text-based information extraction, and putting it on par with precision. IE has been very successful in playing a key role to construct large knowledge bases, with well controlled error rates. The next big challenge now is to overcome the inherent incompleteness of knowledge bases. Understanding and automatically assessing the recall of IE outputs is a vital part in this future research agenda.

### References


IBM. Special issue on "This is Watson". *IBM Journal of Research and Development*, 2012.


Mausam. Open information extraction systems and downstream applications. *AAAI*, 2016.


Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. *ACL*, 2009.


