KnowMore - Knowledge Base Augmentation with Structured Web Markup

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Ran Yu [a,*], Ujwal Gadiraju [a], Besnik Fetahu [a], Oliver Lehmberg [b], Dominique Ritze [b] and Stefan Dietze [a]
[a] *L3S Research Center, Appelstr. 4, 30167 Hannover, Germany*
E-mail: {yu, gadiraju, fetahu, dietze}@l3s.de
[b] *Data and Web Science Group, University of Mannheim, B6, 26 68159 Mannheim, Germany*
E-mail: {oli,dominique}@informatik.uni-mannheim.de

Abstract. Knowledge bases are in wide-spread use for aiding tasks such as information extraction and information retrieval, where Web search is a prominent example. However, knowledge bases are inherently incomplete, particularly with respect to tail entities and properties. On the other hand, embedded entity markup based on Microdata, RDFa, and Microformats have become prevalent on the Web and constitute an unprecedented source of data with significant potential to aid the task of knowledge base augmentation (KBA). However, RDF statements extracted from markup are fundamentally different from traditional knowledge graphs: entity descriptions are flat, facts are highly redundant and of varied quality, and, explicit links are missing despite a vast amount of coreferences. Therefore, data fusion is required in order to facilitate the use of markup data for KBA. We present a novel data fusion approach which addresses these issues through a combination of entity matching and fusion techniques geared towards the specific challenges associated with Web markup. To ensure precise and diverse results, we follow a supervised learning approach based on a novel set of features considering aspects such as quality and relevance of entities, facts and their sources. We perform a thorough evaluation on a subset of the Web Data Commons dataset and show significant potential for augmenting existing KBs. A comparison with existing data fusion baselines demonstrates superior performance of our approach when applied to Web markup data.

Keywords: Keyword one, keyword two, keyword three, keyword four, keyword five

1. Introduction

Knowledge bases (KBs) such as Freebase [3] or YAGO [32] are in wide-spread use to aid a variety of applications and tasks such as Web search and Named Entity Disambiguation (NED). While KBs capture large amounts of factual knowledge, they still are incomplete, i.e. coverage and completeness vary heavily across different types or domains. In particular, long-tail entities and properties are usually insufficiently represented. For instance, considering a selected set of popular predicates used to describe books (Section 3), Freebase is missing respective statements in 63.8% of entities, Wikidata in 60.9% and DBpedia in 49.8%.

Recent efforts aim at exploiting data extracted from the Web to aid knowledge base augmentation (KBA). For instance, Knowledge Vault [7] uses triples extracted from Web documents, while recent works exploit semi-structured data from Web tables [26,27]. The approach described in [36] uses Web search templates. On the other hand, data fusion techniques aim at identifying the most suitable value (or fact) from a given set of observed values, for instance, the correct director of a movie from a set of candidate facts extracted from the Web [8]. To this end, data fusion tech-

[*] Corresponding author. E-mail: yu@l3s.de.

1570-0844/0-1900/$35.00 © 0 – IOS Press and the authors. All rights reserved

niques are fundamental when attempting to solve the KBA problem from observed Web data.

While the extraction of structured data from Web documents is costly and error-prone, the recent emergence of embedded and structured Web page markup has provided an unprecedented source of explicit entity-centric data, describing factual knowledge about entities contained in Web documents. Building on standards such as RDFa[1], Microdata[2] and Microformats[3], and driven by initiatives such as schema.org, a joint effort led by Google, Yahoo!, Bing and Yandex, Web markup has become prevalent on the Web. The Web Data Commons (WDC) [18], an initiative investigating a Web crawl of 2.01 billion HTML pages from over 15 million pay-level-domains (PLDs) found that 30% of all pages contain some form of embedded markup already, resulting in a corpus of 20.48 billion RDF quads[4]. Authors also identify an upward trend of adoption, where the proportion of pages containing markup increased from 5.76% to 30% between 2010 and 2014.

Through its wide availability, markup lends itself as diverse source of input data for the KBA problem. However, the specific charateristics of facts extracted from embedded markup pose particular challenges [37]. Co-references are very frequent (for instance, in the WDC2013 corpus, 18,000 entity descriptions of type `Product` are returned for query `'Iphone 6'`), but are not linked through explicit statements. In contrast to traditional strongly connected RDF graphs, RDF markup statements mostly consist of isolated nodes and small subgraphs. In addition, extracted RDF markup statements are highly redundant and often limited to a small set of highly popular predicates, such as `schema:name`, complemented by a long tail of less frequent statements. Moreover, data extracted from markup contains a wide variety of errors, ranging from typos to the frequent misuse of vocabulary terms[17].

In this work, we introduce *KnowMore*, an approach based on data fusion techniques which enables the exploitation of markup crawled from the Web as diverse source of data to aid KBA. Our approach consists of a two-fold process, where first, candidate facts for augmentation of a particular KB entity are retrieved through a combination of blocking and entity matching techniques. In a second step, correct and novel facts are selected through a supervised classification approach and a novel set of features. We apply our approach to the WDC2015 dataset and demonstrate superior performance compared to state-of-the-art data fusion baselines. In addition, we demonstrate the capability for augmenting three large-scale knowledge bases, namely Wikidata[5], Freebase and DBpedia[6] through markup data based on our data fusion approach. The main contributions of our work are threefold:

- **Pipeline for data fusion on Web markup.** We propose a pipeline for data fusion (Section 3.3) that is tailored to the specific challenges arising from the characeristics of Web markup. Novelty and correctness of facts is addressed through a combination of entity matching, data fusion and diversification techniques. To the best of our knowledge, this is the first approach addressing the task of data fusion on Web markup data.
- **Model & feature set.** We propose a novel data fusion approach consisting of a supervised classification model (Section 5). We propose and evaluate an original set of features which validate correctness of markup facts and use these in a supervised classification model. Experimental results demonstrate high precison (avg. 89.9%) and recall (avg. 77.8%) of our model, outperforming the state-of-art baselines.
- **Knowledge base augementation from markup data.** As part of our experimental evaluation, we demonstrate the use of fused markup data for augmenting three well-established knowledge bases. Our results show a significant potential for addressing the KBA task, where *KnowMore* is able to populate 100% of missing statements for particular properties, for instance, book descriptions in Freebase and Wikidata. On average, *KnowMore* populates 14.7% of missing statements in Wikidata, 11.9% in Freebase and 23.7% in DBpedia. We also investigate the particular potential for augmenting tail entities and properties in Section 9.1.

The paper is structured as follows: Section 2 discusses related work on knowledge base augmentation and

[1] RDFa W3C recommendation: http://www.w3.org/TR/xhtml-rdfa-primer/

[2] http://www.w3.org/TR/microdata

[3] http://microformats.org

[4] http://www.webdatacommons.org

[5] https://www.wikidata.org/

[6] http://dbpedia.org

data fusion, while Section 3 introduces the motivation, problem statement and an overview of our approach. Section 4 and 5 describe the detailed steps for entity matching, data fusion and diversification, while Section 6 describes the experiment setup followed by the results in Section 7. We assess the potential to generalise our supervised approach across distinct types in Section 8 and provide a thorough discussion of the KBA potential of markup as well as limitations of our work in Section 9 and conclude by proposing future work (Section 10).

2. Related Work

In this section we review related literature. We focus on two main lines of work on Linked Data, *knowledge-base augmentation* and *data fusion* as the most closely related fields to our work.

2.1. Knowledge-base Augmentation (KBA)

The main goal of KBA is to discover facts pertaining to entities and augmenting Knowledge Bases (KB) with these facts [35,12].

Some previous works have proposed approaches that suggest augmenting KBs with internal data. Such works typically focus on predicting the type [15] of an entity or finding new relations based on existing data [5,6,30]. Other prior works are more closely relevant to our problem setup; in that they focus on predicting relations with external data. Notable works propose the use of Wikipedia as a text corpus annotated with entities, seek for patterns based on existing KB relations, and further apply the patterns to find additional relations for DBpedia [1] or Freebase [21]. News corpora have also been used for augmenting DBpedia with similar settings [11]. Paulheim et al. [22] proposed to identify common patterns of instances in the Wikipedia list page and apply the patterns to add relations to the remaining entities in the list. Dong et al. proposed '*Knowledge Vault*' [7], a framework for extracting triples from webpages and aims at constructing a KB from Web data. Dutta et al. [10] focus on the mapping of relational phrases such as facts extracted by '*Nell*' and '*Reverb*' to KB properties. Furthermore, they group the same semantic relationships represented by different surface forms together through Markov clustering. Recent works by Ritze et al. use relational HTML tables available on the Web to fill missing values in DBpedia [26,27]. The authors propose to first match the tables to the DBpedia entities, and then compare several data fusion strategies such as voting and the Knowledge-Base Trust (KBT) score to identify valid facts.

Ristoski et al. proposed an approach to enrich product ads with data extracted from Web Data Commons [25]. The approach extracts attribute-value pairs from plain text and matches them to database entities with supervised classification models. The notable methods described in previous works are tailored to specific data sources, which have different characteristics compared to markup data. Hence, merely adopting the existing methods to cater for markup data is not sufficient. However, we have revised and adopted some of the features in our proposed approach.

Other works suggest using the whole Web as a potential data corpus through search engines [13,36]. QA-based approaches are often designed to facilitate the filling of values of a specific set of properties, and they rely on manually created templates. This limits their application to constrained sets of properties.

Existing works typically assume that there is only one true value for a property when resolving conflicts. In contrast, we focus on both diversity and completeness by catering for multiple correct values, as multiple-cardinality properties are prevalent. Another limitation of existing KBA works is that the novelty of the discovered facts is ignored; there is an overlap between the result and the facts existing in a KB. On the contrary, our approach aims at providing correct and novel results that can be of immediate value to the KB.

2.2. Data Fusion

Data fusion is defined as "the process of fusing multiple records representing the same real-world object into a single, consistent, and clean representation" [2]. In the context of the Semantic Web, previous works on data fusion can be classified into two classes – *heuristic-based* and *probability-based*.

Heuristic-based Methods. Schultz et al. introduced '*LDIF*', that uses user provided heuristics to find duplicate real-world entities [29]. Mendes et al. proposed '*Sieve*', which resolves conflicts in Linked Data from different sources by selecting one value for each property based on quality measures such as recency and frequency [16,4]. '*ODCleanStore*' provides heuristic-based mechanisms such as max frequency for resolving conflicts during the fusion of Linked Data [14,20]. One of the limitations of such heuristics-based approaches is that they rely on the observation of a specific dataset, which is often not generalizable for other datasets. Furthermore, the heuristics usually focus on a single aspect of the quality, e.g. recency or frequency, while the quality of a resource is typically influenced by multiple factors to varying degrees.

Probability-based Methods. Zhao et al. [39] proposed an unsupervised probabilistic graphical model to infer true records and source quality based on the false-positives and false-negatives of the data source. In [8], Dong et al. introduced data fusion techniques which identify true subject-predicate-object triples, that are extracted by multiple extractors and originate from multiple sources [7]. In this work, authors selected and adapted three existing data fusion techniques and improved them with a series of refinements. Furthermore, Pochampally et al. proposed to use joint precision and joint recall to indicate correlation between sources in order to penalize the copying between sources [24]. In later work, the authors proposed a probabilistic model to compute the Knowledge-Based Trust (KBT), i.e., a score for measuring the trustworthiness of the resources [9]. KBT focuses on the general quality of a resource, and is computed based on the relation between a resource and Freebase.

The major difference between our work and the aforementioned works is that previous works focus only on the *correctness* by measuring the quality of the source. In contrast, we not only consider the source quality but also the features of predicate and object of a fact. Furthermore, our data fusion approach can better adapt to dynamic data as it does not need to refuse the entire dataset when new instances are added.

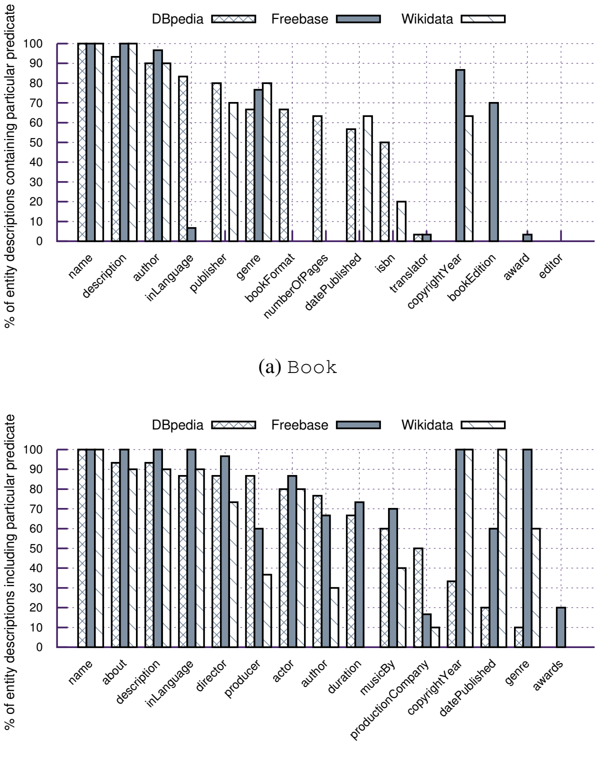
Our recently published work presents an entity summarization approach that retrieves entities from WDC and selects facts to build diversified entity descriptions based on clustering [38]. While the main focus of our previous work was diversification, in this paper we focus on both precision and diversity of the entity description.

3. Motivation & Approach

3.1. Motivation

Type-specific investigations [28,33,37] have shown the complementary nature of markup data, when compared to traditional knowledge bases, where the extent of additional information varies strongly between types.

For a preliminary analysis of *DBpedia*, *Freebase* and *Wikidata*, we randomly selected 30 Wikipedia entities type *Movie* and *Book* and retrieve the corresponding entity descriptions from all three KBs. We select the 15 most frequently populated properties for each type and provide equivalence mappings across all KB schemas as well as the *schema.org* vocabulary manually[7]. Since all vocabulary terms and types in the following refer to *schema.org*, prefixes are omitted. Figure 1 shows the proportion of instances for which the respective property is populated for the movie case. Indicated properties refer to the corresponding *schema.org* term. There is a large amount of missing facts across all KBs for most of the properties, with an average proportion of missing statements for books (movies) of 49.8% (37.1%) for DBpedia, 63.8% (23.3%) for Freebase and 60.9 % (40%) for Wikidata.



(a) `Book`

(b) `Movie`

Fig. 1. Proportion of book entity descriptions per KB that include particular predicates.

[7] The mappings are online at: `http://l3s.de/~yu/knowmore/`

In addition, coverage varies heavily across different properties, with properties such as *editor* or *translator* being hardly present in any of the KBs.

On the other hand, tail entities/types as well as highly dynamic properties which require frequent updates, such as the price tag of a product are prevalent in markup data [19], yet tend to be underrepresented in structured KBs. Hence, markup data lends itself as data source for the KBA task. However, given the specific characteristics of markup data [37], namely the large amount of coreferences and near-duplicates, the lack of links and the variety of errors, data fusion techniques are required which are tailored to the specific task of KBA from Web markup.

3.2. Problem Definition

We consider data accumulated by extracting structured Web markup from Web documents which is stored in n-quad format. We refer to such a dataset as M, where the WDC dataset is an example. In M, each entity description corresponds to a set of $\langle s, p, o, u \rangle$ quadruples, where s, p, o, u represent subject, predicate, object and the URL of the document from which the triple has been extracted respectively. For a particular real-world entity e, usually there exist $n \geq 0$ subjects s from the quadruples $\langle s, p, o, u \rangle$ which represent distinct descriptions of e. We define $e_s = \langle s, p_i, o_i \rangle$ as the entity description of e corresponding to subject s.

As input for the KBA task we consider an entity description e_q from a given KB representing a real-world entity q. We define the KBA task from a given markup corpus M given a knowledge base entity description e_q as follows:

Definition 1 *KBA: For an entity q that has a corresponding description e_q in a KB we aim at selecting a set of facts $f_i \in F'$ from M which augment the KB description e_q for q. Each fact f_i represents a valid property-value pair $\langle p_i, o_i \rangle$ describing the entity q.*

We consider a fact valid for augmentation, if it meets the following criteria:

- A fact is *correct*, i.e. consistent with the real world regarding query entity q
- A fact presents *novel* information with regard to the entity description e_q of q in a given KB.
- The predicate p_i of fact $\langle p_i, o_i \rangle$ should already be reflected in a KBs given schema.

As defined in the last statement, we limit ourselves to the augmentation of existing predicates for the sake of this work. However, we include a detailed discussion of the augmentation of statements involving new properties relative to a given KB in Section 9.

3.3. Approach Overview

Our approach (*KnowMore*) for addressing the KBA problem defined above, consists of two steps, namely (i) entity matching, and (ii) data fusion. We introduce the intuition behind each step below.

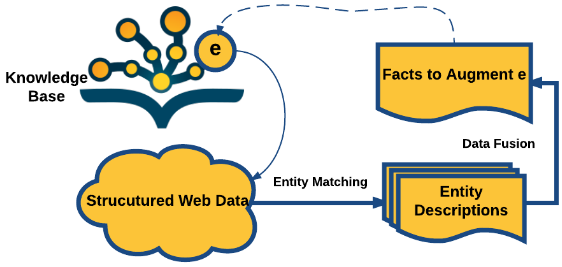


Fig. 2. Overview of pipeline.

Entity matching. The first step aims at obtaining candidate facts by collecting the set of co-referring entity descriptions $e_i \in E$ from M which describe q and corefer to the entity description e_q in a given KB. We use a three step approach in order to efficiently achieve high accuracy results.

- Blocking with standard BM25 entity retrieval on the *name* field to reduce the search space.
- Data cleansing to improve general data quality.
- Validation based on a weighted similarity between e_q and the entity descriptions in E.

Hence, we retrieve the set E containing candidate entity descriptions represented through facts $f \in F$ that potentially describe q.

Data fusion. During the data fusion step, we aim at selecting a subset $F' \subset F$ that fulfills the attributes as listed in 3.2. More specifically, we introduce *data fusion* techniques (Section 5) based on supervised classification to ensure the correctness (Section 5.1), and apply a diversification step to ensure novelty (Section 5.2).

We describe each step of the approach in the following two sections in detail.

4. Entity Matching

The entity matching step ($KnowMore_{match}$) aims at detecting a set of candidate entity descriptions $e_i \in$

E with $E \subset M$ which are likely to be coreferences of a given KB entity description e_q. We apply three steps in $KnowMore_{match}$, namely blocking, cleansing and matching. These steps are applied iteratively over all entities which are to be augmented.

4.1. Blocking

We apply the standard BM25 entity retrieval approach as a blocking technique to reduce the search space. We created an index for each type-specific subset using Lucene, and then use the *label* of e_q to query the field *name* within a type-specific index. Hence, queries for a specific type/label-combination which represents e_q result in a set of candidate entity descriptions $e_i^0 \in E_0$ that potentially describe the same entity as e_q. [34] shows that string comparison between labels of markup entities is an efficient way for obtaining potential co-references.

For instance, considering the query "*Brideshead Revisited*" (of type *Book*), as part of the the blocking step we query the Lucene index after resolving object properties and obtain 1,657 entity descriptions consisting of 15,940 quads. An excerpt of the result set is shown in Table 1.

Table 1
Excerpt from the result set (1,657 entity descriptions in total) for the query "*Brideshead Revisited*" (of type *Book*) after blocking.

Node ID	Property	Value
_:node1	author	Evelyn Waugh
_:node1	datePublished	1940
_:node1	isbn	9781904605577
_:node2	author	Waugh, Evelyn
_:node2	publisher	Back Bay Books.
_:node3	author	Roger Parsley
_:node3	publisher	Samuel French Ltd

4.2. Data Cleansing

Based on earlier work [17], which studied common errors in semantic markup, we implement heuristics described in this work and apply these to E as a cleansing step thereby improving the quality of the data. In particular, we implemented the heuristics as proposed in [17] to:

– **Fix wrong namespaces.** Most namespace issues seem due to typing errors, e.g. lacking a slash or using *https://* instead of *http://.* Another reason is the misuse of the upper/lower-cases in a case-sensitive context, e.g. use of *Schema.org* instead of the valid term *schema.org*.

– **Handle undefined types and properties.** The use of undefined types and properties are frequent in Web markup data. Some of the undefined types exist due to typos and the misuse of the upper/lower-cases, e.g. the use of *creativework* for the intended type *CreativeWork*, where simple heuristic can be applied to resolve these issues.

Applying these heuristics improves the performance of the subsequent step by providing a wider and improved pool of candidates.

4.3. Entity Matching

As studied by Meusel et al. [19], resolving coreferences simply through pseudo-key properties does not produce sufficient results when applied on Web markup data. Thus, we adapt the entity matching approach described in [25] to filter out noise in E_0, for instance, entity descriptions which are not relevant to q but fetched through the initial blocking step due to ambiguous labels. Our matching approach builds on the assumption that the importance of distinct properties differs when computing entity similarity, with pseudo-key properties being the most decisive ones. For example, instances of type *Book* which share the same *author* have a higher probability to be equivalent than books which share the same *bookFormat*.

In order to compute the similarity for each property, we consider all properties as attributes of the feature space $\overrightarrow{A} = \{a_1, a_2, ..., a_n\}$, so that each entity description e can be represented as a vector of facts $\overrightarrow{v} = \{f_{a1}, f_{a_2}, ..., f_{a_n}\}$. We construct a similarity vector $\overrightarrow{Sim}(\overrightarrow{v^{KB}}, \overrightarrow{v})$ between e_q and each entity description $e_i^0 \in E_0$ as in Equation 1.

$$\overrightarrow{Sim}(\overrightarrow{v^{KB}}, \overrightarrow{v}) = \{\lambda_{a_1}, \lambda_{a_2}, ..., \lambda_{a_n}\} \qquad (1)$$

$$\lambda_{a_i} = Sim(f_{a_i}^{KB}, f_{a_i}) \qquad (2)$$

In order to compute $Sim(f_{a_i}^{KB}, f_{a_i})$, we deploy similarity metrics f which are datatype-specific, that is, we implemented one similarity measure for each *schema.org* datatype, and automatically select the appropriate metric. For instance, for text/literals, we deploy cosine similarity. The source code of the similarity metrics can be found online[8].

We then train a supervised classification model, to make the decision whether or not e_i^0 is a match for e_q. We experimented with several state-of-the-art classifiers (SVM, kNN with varying ks and Naive Bayes). Since Naive Bayes achieves a precision that is 14% higher than the best SVM (linear kernel), and 23% higher than the best KNN ($k = 3$), we rely on a trained Naive Bayes classifier. The classification and clustering implementation in our approach is built on top of the Java-ML toolkit[9]. The training data is described in Section 6.2. This step results in the set of co-referring entity descriptions $e_i \in E$ which provide the candidate facts $f_i \in F$ for the following steps.

Referring to our example, after removing the unmatched entity descriptions through the entity matching step from the blocking result, there are 44 matched entity descriptions remaining in the result set. Some examples are shown in Table 2, where, for instance, *_:node3* had been removed since it refers to the stage play rather than the book and does not match entity q.

Table 2
Excerpt from result set (44 entity descriptions in total) for the query, "*Brideshead Revisited*" (of type *Book*) after entity matching.

Node ID	**Property**	**Value**
_:node1	author	Evelyn Waugh
_:node1	datePublished	1940
_:node1	isbn	9781904605577
_:node2	author	Waugh, Evelyn
_:node2	publisher	Back Bay Books.

5. Data Fusion

This step aims at fusing candidate entity descriptions in E by detecting the correct and novel facts $f' \in F'$ with $F' \subset F$ to augment e_q.

5.1. Correctness - Supervised Classification

The first step ($KnowMore_{class}$) aims at detecting correct facts by learning a supervised model that produces a binary classification for a given fact $f \in F$ into one of the labels {*'correct'*, *'incorrect'*}. We rely on a Naive Bayes classifier since our experiments have shown superior performance over SVM with a precision increase of 10.01% compared to SVM (linear kernel).

We introduce the features used for our supervised learning approach in Table 3 and describe them in detail below. Through an initial data analysis step, all features have been identified as potential indicators of fact correctness.

While we aim to detect the correctness of a fact, we consider characteristics of the *source*, that is the Pay-Level-Domain (PLD) from which a fact originates, the *entity description*, the *predicate* term as well as the *fact* itself. The four different categories are described below.

Source level. As has been widely studied in previous works, source quality is an important indicator for data fusion [39,24,9]. Features $t_i^r, i \in [1,3]$ consider the PageRank score as an authority indicator of the PLD from which a fact is extracted, assuming a higher PageRank indicates higher authority and hence quality. The intuition behind features $t_i^r, i \in [4,9]$ is that, there is a higher potential of a PLD to provide incorrect facts if more errors have been detected across markup from a respective PLD. We consider the rate of common errors detected based on previously identified heuristics [17] to compute $t_i^r, i \in [4,6]$ and the precision of a PLD computed based on our ground truth (Section 6.2) as quality indicators and hence extract feature $t_i^r, i \in [7,9]$.

Entity level. Based on the data analysis, entity descriptions containing a large number of facts are usually of higher quality. Thus, we use the size of entity descriptions, reflected through features $t_i^e, i \in [1,3]$, as additional indicator of quality.

Property level. The quality of facts strongly varies across predicates, as identified in previous studies [37, 19], with some properties being more likely to be part of a correct fact than others. One example of a predicate often included in incorrect statements is *datePublished* of a movie, that is often mistakenly used to describe the publishing time of the Web document. Following this observation, we extract features $t_i^p, i \in [1,5]$ to consider characteristics of the involved predicate terms, such as their frequency.

[8] http://l3s.de/~yu/knowmore/
[9] http://java-ml.sourceforge.net/

Table 3
Features for supervised data fusion from markup data.

Category	Notation	Feature description
Source level	t^r_1, t^r_2, t^r_3	Maximum, minimum, average PageRank score of the PLDs containing fact f
	t^r_4, t^r_5, t^r_6	Maximum, minimum, average percentage of common errors [17] of the PLDs containing fact f
	t^r_7, t^r_8, t^r_9	Maximum, minimum, average precision (based on training data) of the PLDs containing fact f
Entity level	t^e_1, t^e_2, t^e_3	Maximum, minimum, average size (number of facts) of e_s containing f
Property level	t^p_1	Predicate term
	t^p_2	Predicate frequency in F
	$\dagger t^p_3$	Amount of clusters of predicate p
	$\dagger t^p_4$	Average cluster size of predicate p
	$\dagger t^p_5$	Variance of the cluster sizes of predicate p
Fact level	t^f_1	Fact frequency in F
	$\dagger t^f_2$	Normalized cluster size that f belongs to

†-features extracted based on clustering result

Given that our candidate set contains vast amounts of near-duplicate facts, often using varied surface terms for the same or overlapping meanings, we approach this problem through clustering of facts. We employ the X-Means algorithm [23], as it is able to automatically determine the number of clusters. This clustering step aims at grouping or canonicalizing different literals or surface forms for specific object values. For instance, *Tom Hanks* and *T. Hanks* are equivalent surface forms representing the same entity. To detect duplicates and near-duplicates, we first cluster facts that have the same predicate p into n clusters $(c_1, c_2, \cdots, c_n) \in C$. In this way, considering string similarity, we can canonicalize equivalent surface forms. The performance of the clustering on removing near duplicates is discussed in Section 7.3. Another challenge considered here is the cardinality of predicates. Depending on the predicate, the number of potentially correct statements varies. For example, *actor* is associated with multiple values, whereas *duration* normally has only one valid statement. This is reflected in the cluster amount n for a given predicate (t^p_3). The intuition behind feature t^p_4 is that the average size of clusters is a indicator of the frequency of facts in p which usually correlates with the quality. Feature t^p_5 is extracted based on the observation that in most cases, wrong facts have lower frequency than average, thus the variance is larger if there are wrong facts among the facts of p.

Fact level. Fact frequency [16] has been used in previous data fusion works and is shown to provide efficient features for determining the correctness of facts. Based on these insights, we extract features $t^f_i, i \in [1, 3]$. We consider the size of a cluster as feature t^f_2 indicating the frequency of a fact, where the normalized size of cluster c_i is $|c_i| / \sum_{j=1}^{n} |c_j|$.

From the computed features we train the classifier for classifying the facts from F into the binary labels $\{$*'correct'*, *'incorrect'*$\}$. More details about the training and evaluation through 10-fold cross-validation are presented in Section 7.2. The *'correct'* facts form a set F'_{class} that is the input for next steps.

Referring to our running example, after removing wrong facts from the candidate facts, such as *datePublished: 1940* through the classification step, we obtain 37 correct facts in the result set for the query *Brideshead Revisited, type:(Book).* An excerpt of the resulting facts are shown in Table 4.

5.2. Novelty

A fact f is considered to be *novel* with respect to the KBA task, if it fulfills the conditions: i) not duplicate with other facts selected from our source markup

Table 4
Excerpt from result set (37 distinct correct facts) for query "*Brideshead Revisited*" (of type *Book*) for $KnowMore_{class}$.

Class	Property	Value
correct	s:author	Evelyn Waugh
correct	s:isbn	9781904605577
correct	s:author	Waugh, Evelyn
incorrect	datePublished	1940
correct	s:publisher	Back Bay Books.

corpus M, ii) not duplicate with any facts existing in the KB. Each of these two conditions corresponds to a diversification step.

Diversification with respect to M ($KnowMore_{div}$*).* As introduced in Section 5.1, we detect near-duplicates via clustering. For each predicate p, all the facts $f = \langle p, o_i \rangle$ corresponding to p are clustered into n clusters $\{c_1, c_2, \cdots, c_n\}$. Each cluster $c_i, i = 1, ..., n$ contains a set of near-duplicates. To fulfill i), we select only one fact from each cluster by choosing the fact that is closer to the cluster's centroid. This results in the fact set F'_{div} that is the input for next diversification step.

Diversification with respect to KB ($KnowMore_{nov}$*).* We compute the similarity $Sim(f_i, f_{KB})$ between facts f_{KB} in a respective KB for a particular predicate p and facts f_i for the same (mapped) predicate p in F'_{div} with the datatype-specific similarity metrics as introduced in Section 4. If $Sim(f_i, f_{KB})$ is higher than a threshold τ, we remove the fact along with its near-duplicates, i.e. the facts in the same cluster from the candidate set F'. We explain τ and its configuration during the experimental Section 6.3. The facts selected from F' in this step are the final result for augmenting the KB. Referring to the example, the fact *author: Waugh, Evelyn* is removed during the diversification with regard to M as it is a duplicate of fact *author: Evelyn Waugh*, yet has been selected as more representative.

With respect to the KBA task, consider the augmentation of the example entity "*Brideshead Revisited*" (of type *Book*) as illustrated in Table 5. The example facts #2 and #3, would be valid results of the KBA task for DBpedia since they are **novel**, while only fact #2 is a valid augmentation for Freebase and Wikidata as it is the only fact that is novel.

6. Experimental Setup

Here we describe the experimental setup for our evaluation.

Table 5
Novelty of correct, distinct facts with regard to KBs for the query "*Brideshead Revisited*" (of type *Book*).

ID	Fact	DBpedia	Wikidata	Freebase
1	author, Evelyn Waugh	✗	✗	✗
2	isbn, 9781904605577	✔	✔	✔
3	publ., Back Bay Books	✔	✗	✗

6.1. Data

Dataset We use the WDC2015 dataset[10], where we extracted 2 type-specific subsets consisting of entity descriptions of the *schema.org* types *Movie* and *Book*. Initial experiments indicated that these types are well reflected in the WDC2015 datasets, and at the same time, their facts are comparably easy to validate manually when attempting to label a ground truth. The *Movie* subset consists of 116,587,788 quads that correspond to 23,334,680 subjects/nodes, and the *Book* subset consist of 174,459,305 quads and 34,655,078 subjects.

Entities & KBs to Augment As input for the KBA task, we randomly select 30 entities for each type *Book* and *Movie* from Wikipedia. We evaluate the performance of our approach for augmenting entity descriptions of these 60 entities obtained from three different KBs: DBpedia (*DB*), Freebase (*FB*) and Wikidata (*WD*). For DBpedia, we retrieve entity descriptions through the SPARQL endpoint[11] where resource URIs were obtained by replacing the Wikipedia namespace of our selected entities with the DBpedia resource path. URIs of corresponding Freebase and Wikidata entity descriptions are obtained through the *owl:sameAs* links present in DBpedia. Using these URIs, the respective entity descriptions are obtained through the latest available version of Freebase[12] (accessed Sep 30, 2016) and the Wikidata SPARQL endpoint[13].

The full list of entities can be found online[8]. A preliminary analysis of the completeness of these obtained entity descriptions is shown in Figure 1 in Section 3.1.

[10] http://webdatacommons.org/structureddata/index.html#toc3

[11] http://dbpedia.org/sparql

[12] http://commondatastorage.googleapis.com/freebase-public/rdf/freebase-rdf-latest.gz

[13] https://query.wikidata.org/sparql/

Properties to Augment To simplify the schema mapping problem between WDC data and the respective KBs while at the same time, taking advantage of the large-scale data available in our corpus, we limit the task to entities annotated with the *http://schema.org* ontology for this experiment. Previous works have shown that schema.org is the only vocabulary which is consistently used at scale [19]. We manually create a set of schema mappings that maps the *schema.org* vocabularies to the *DB, FB, WD* vocabularies. For this, we first select all the *schema.org* predicates appearing in F. We identify the ones that have equivalent properties within all involved vocabularies and create equivalence mappings (*owl:equivalentProperty*). The list of predicates and the mapping statements can be found online[8].

6.2. Ground Truth & Metrics

6.2.1. Ground Truth via Crowdsourcing

Entity Matching. We used crowdsourcing to build a ground truth by acquiring labels for each $e_i \in E$. In each case, crowd workers were presented with the entity description e_q, i.e. the Wikipedia page, and entity description $e_i \in E$, and were asked to validate e_i as either *valid*, *invalid* or *insufficient information to judge* with respect to e_q. We deployed the task on CrowdFlower[14], and gathered 5 judgments from distinct workers on each (q, $e_i \in E$) pair. To ensure high quality, we restricted the participation to Level 3 workers alone[15]. In addition to this, we used test questions to flag and reject untrustworthy workers. Workers were compensated at the rate of 6 USD cents per judgment. On average, workers performed with an accuracy of 92% on the test questions, indicating high reliability. The inter-rater agreement between workers was 89% using pairwise percent agreement (PPA). By applying this process on E_0, we obtain 89 (180) *valid* and 128 (118) *invalid* entity descriptions for *Movie* (*Book*) entities respectively.

Data Fusion - Correctness. Similarly, we used crowdsourcing to build a ground truth for the correctness of facts $f_i \in F$. For the valid entity descriptions in E, we acquire labels for all distinct facts, as either *correct* or *incorrect* with respect to q. We acquired 5 judgments from distinct workers for each entity and corresponding facts through Crowdflower. We used similar quality control mechanisms as in the entity matching task. Workers were compensated at the rate of 6 USD cents per judgment. Workers performed with an accuracy of 95% on the test questions. The inter-rater agreement between workers was 86.9% using pairwise percent agreement (PPA). This indicates a high reliability of the ground truth. This process results in 371 (out of 456) and 298 (out of 341) *correct* facts for *Movie* and *Book* dataset respectively. Distinct facts were obtained by removing duplicate literals, null values, URLs and the unresolved objects (e.g. *node3* that could not be resolved in the dataset). The ground truth is publicly available [8].

Data Fusion - Novelty. We built corresponding ground truths for validating (i) diversity within M, as well as (ii) novelty with respect to the different *KBs*. Three authors of this paper acted as experts and designed a coding frame to decide whether or not a fact is novel. After resolving disagreements on the coding frame on a subset of the data, every fact was associated with one expert label through manual deliberation. We followed the guidelines laid out by Strauss [31] during the coding process. Distinct facts were obtained by removing duplicate literals, null values, URLs and unresolved objects.

6.2.2. Metrics

We consider distinct metrics for evaluating each step of our approach.

- *Entity matching.* Precision P - the percentage of entity descriptions $e_i \in E$ that were correctly matched to e_q, R - the percentage of $e_i \in E_0$ that were correctly matched to KB, and the $F1$ score.
- *Correctness.* We evaluate the performance of the approaches through standard precision P, recall R and $F1$ scores, based on our ground truth.
- *Novelty.* We evaluate the performance of both substeps: i) diversification with respect to M ($KnowMore_{div}$), ii) diversification with respect to a given KB ($KnowMore_{nov}$). For i), we compute $Dist\%$ - the percentage of distinct facts within the respective result set. We compare between $Dist\%$ (F'_{div}) and $Dist\%$ (F'_{class}), that is, before and after the diversification within M. For (ii), we measure the novelty as Nov - the percentage of novel facts - and compare between Nov (F'_{div}) and Nov (F'), that is, the novelty before and after this step.

[14] http:www.crowdflower.com/

[15] Level 3 workers on CrowdFlower have the best reputation and near perfect accuracy in hundreds of previous tasks.

6.3. Configuration & Baselines

Configuration. We deploy our approach as described in Section 4.3 and 5. We experimentally identified the best value of threshold $\tau = 0.5$.

Baselines. We consider 2 different baselines and compare (*KnowMore$_{class}$*) with *PrecRecCorr* that is proposed by Pochampally et al. [24] and *CBFS* [38]. To the best of our knowledge, the *CBFS* approach is the only available method so far directly geared towards the challenges of markup data, while *PrecRecCorr* represents a recent and highly related data fusion baseline.

– *PrecRecCorr*@k: facts selected based on the approach from candidate set F. We consider each PLD as a source and implemented the *exact solution* as described in the paper. We use the threshold as presented in the paper, i.e. 0.5, to classify facts.

– *CBFS*: facts selected based on the *CBFS* approach from F. The *CBFS* approach clusters the associated values at the predicate level into n clusters $(c_1, c_2, \cdots, c_n) \in C$. Facts that are closest to the cluster's centroid of each cluster are select, provided they meet the following criteria:

$$|c_j| > \beta \cdot \max(|c_k|), c_k \in C \quad (3)$$

where $|c_j|$ denotes the size of cluster c_j, and β is a parameter used to adjust the number of facts. In our experiment, β is empirically set to 0.5, which is the same as the best-performing setup as defined in the original paper.

7. Evaluation Results

In this section, we present experimental results obtained through the setup described in the previous sections.

7.1. Entity Matching

While the entity matching step (*KnowMore$_{match}$*) is a precondition for the subsequent fusion step, we provide evaluation results for this step and compare it to entity descriptions obtained through BM25@k as baseline. Since we obtain the corresponding URIs of Freebase and Wikidata entities through the *sameAs* link in DBpedia, here we present only the result of matching entity descriptions to DBpedia. Table 6 shows the evaluation results of the standard precision P, recall R and $F1$ scores.

Table 6

Performance of *KnowMore$_{match}$* and baselines.

	Movie			Book		
Approach	**P**	**R**	**F1**	**P**	**R**	**F1**
KnowMore$_{\text{match}}$	0.943	0.742	**0.83**	0.88	0.894	**0.887**
BM25@20	0.592	0.831	0.692	0.219	0.722	0.336
BM25@50	0.406	1.000	0.578	0.124	1.000	0.220

As presented in Table 6, our supervised matching approach achieves high $F1$ scores of 0.83 and 0.887 respectively, thereby outperforming the BM25@20 and BM25@50 baselines and providing a sound set of candidates for the subsequent step.

7.2. Correctness - Data Fusion

The result of *KnowMore$_{class}$* are shown in Table 7.

Table 7

Performance of *KnowMore$_{class}$* and baselines.

	Movie			Book		
Approach	**P**	**R**	**F1**	**P**	**R**	**F1**
KnowMore$_{\text{class}}$	**0.967**	**0.88**	**0.921**	0.886	**0.874**	**0.88**
PrecRecCorr	0.924	0.861	0.891	**0.893**	0.48	0.624
CBFS	0.868	0.740	0.799	0.849	0.834	0.842

The presented F1 score of the *PrecRecCorr* baseline is the best possible configuration for our given task, where we experimented with different decision making parameters ([0,1], gap 0.1) as discussed in [24] and identified 0.5 experimentally as the best possible configuration. We observe that the $F1$ score of our approach is 25.6% higher than *PrecRecCorr* and 3.8% higher than *CBFS* on average across datasets. This indicates that our approach provides the most efficient balance between precision and recall across the investigated datasets, when applied to the novel task of data fusion from Web markup. Although, the precision of the baseline approach *PrecRecCorr* is insignificantly (0.7%) higher than the one from *KnowMore$_{class}$* on the *Book* dataset, the baseline fails to recall a large amount of correct facts, where the recall of *KnowMore$_{class}$* is approximately 39.4% higher. This also is reflected in the average size of entity descriptions obtained through both approaches, where the entity descriptions from *PrecRecCorr* consist of 4.88 statements on average, and the ones from *KnowMore$_{class}$* are 8.83, indicating a significantly larger potential for the KBA task. A more detailed discussion of the potential impact on the KBA task is provided in Section 9, investigating the KBA potential beyond the narrow definition of the investigated task of this setup, e.g. by augmenting additional predicates not already foreseen in a given KB schema or to populate KBs with additional entities.

7.3. Novelty

This section presents the evaluation results for the diversification steps introduced in Section 5.2.

Diversity. Table 10 presents the evaluation result before ($Dist\%$ (F'_{class})) and after ($Dist\%$ (F'_{div})) the step $KnowMore_{div}$.

Table 8

Diversity *Dist*% before and after diversification.

Dataset	Dist%($\mathbf{F'_{class}}$)	Dist%($\mathbf{F'_{div}}$)
Movie	94.8	**96.1**
Book	82.1	**95.6**

The novelty of facts improves 1.3% for the *Movie* dataset and 13.5% for the *Book* dataset (Table 10). The less significant improvement gain for the *Movie* dataset presumably is due to the nature of the randomly selected *Movie* entities. As these appear to be mostly tail entities, candidate facts in our markup corpus M are fewer and less redundant. Hence, the amount of duplicates and near-duplicates is smaller, reducing the effect of the diversification. Diversification is of particular importance for popular entities and well-represented entities.

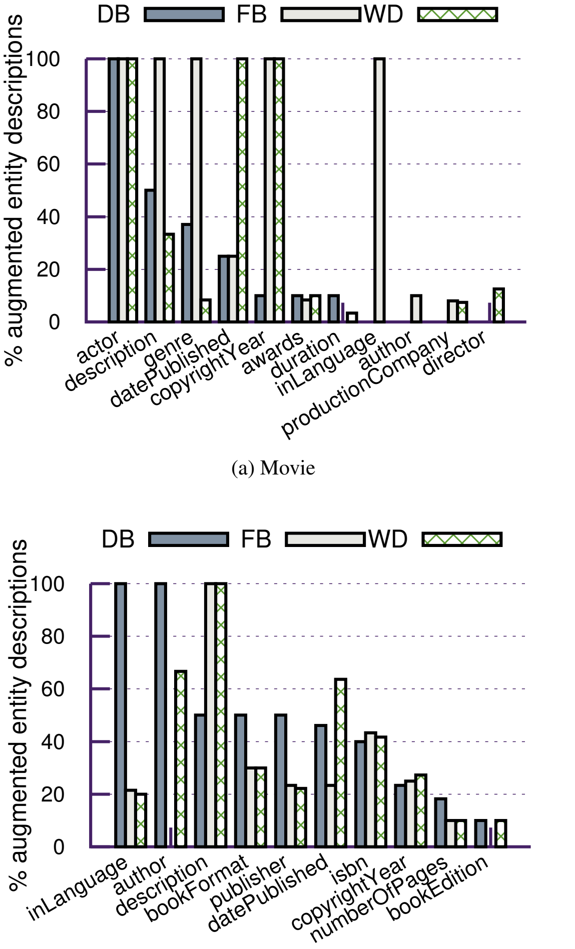
Novelty with respect to KB. The results before (Nov (F'_{div}))and after (Nov (F')) the diversification for specific KBs are presented in Table 9.

Table 9

Novelty of F_{div} and F' with respect to target KBs.

	Movie		**Book**	
KB	Nov($\mathbf{F'_{div}}$)	Nov($\mathbf{F'}$)	Nov($\mathbf{F'_{div}}$)	Nov($\mathbf{F'}$)
DBpedia	0.631	**0.964**	0.736	**0.948**
Freebase	0.527	**0.739**	0.639	**0.928**
Wikidata	0.412	**0.9**	0.705	**0.955**

This step improves novelty by 29.7% on average across datasets and KBs, by reducing the amount of facts from 254 to 180 for DBpedia, to 175 for Freebase and to 149 for Wikidata. Our final result F' shows a novelty of 90% in most cases, what translates to a minor amount of near-duplicates and a sufficient novelty for augmenting the target KBs.



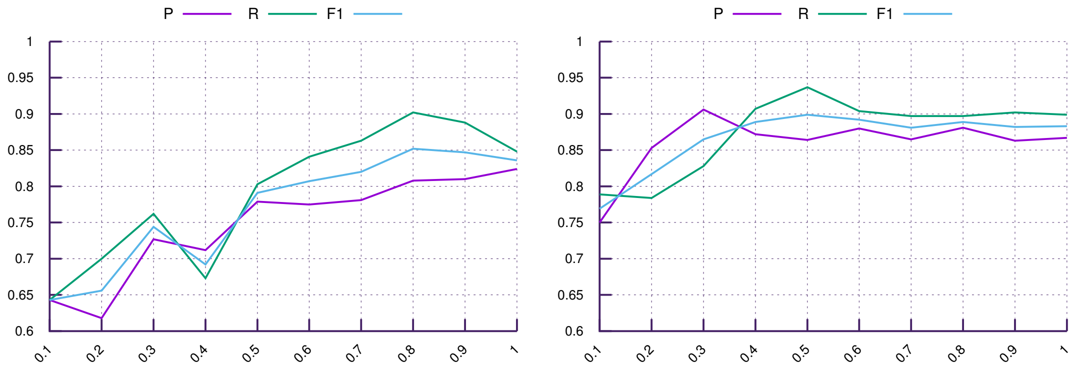
(a) Movie

(b) Book

Fig. 3. Proportion of augmented entity descriptions with *KnowMore*. Only predicates which were augmented in at least one KB are shown.

7.4. Coverage Gain

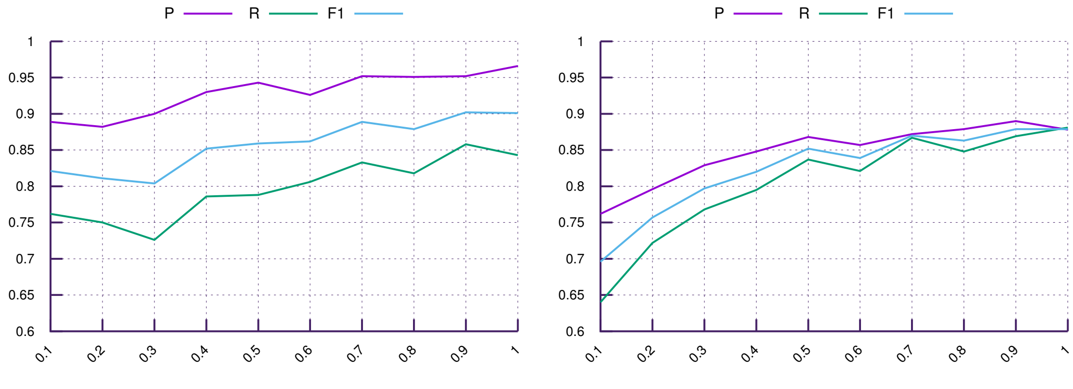
On average across all KBs, more than 50% of the facts obtained in F_{div} are novel with respect to a given KB. Figure 3 shows the percentage of filled slots (of the previously empty ones) per predicate and KB for our selected 30 entities (per type). For instance, within the *Movie* case, for property *actor* we were able to populate 100% of the missing facts in both DBpedia and Freebase. We observe that the obtained gain varies strongly between predicates and entity types, with a generally higher gain for book-related facts.



(a) *Movie*

(b) *Book*

Fig. 4. P, R and F1 score using different size of the training data for $KnowMore_{match}$. X-axis shows the percent of training data, Y-axis shows the P/R/F1 value.



(a) *Movie*

(b) *Book*

Fig. 5. P, R and F1 score using different size of training data for $KnowMore_{class}$. X-axis shows the percent of training data, Y-axis shows P/R/F1 value.

8. Evaluation of Generalisation Potential

As introduced earlier, our approach has been trained on two specific types (*Book* and *Movie*). The intuition behind this choice is that (i) different properties have varied contribution on different types when computing the similarity between entity descriptions in the entity matching step and (ii) particular features such as predicate term t_1^p increase the feature space when introducing new types and associated properties. To this end, we have restricted our experiments to particular types to reduce the required training data. However, given the type-agnostic nature of most features, it seems reasonable to anticipate comparable performance even when applying our approach across types. In this section, we evaluate the generalisation of the *KnowMore* approach with respect to two aspects: 1) the scale of training data required for the supervised fusion step, 2) the performance of our approach when trained with cross-type training data, as opposed to type-specific sets.

8.1. Scale of required Training Data

As described in Section 6.2, our ground truth consists of judgments for 217 (298) entity descriptions for *Movie* (*Book*) entities for the entity matching step $KnowMore_{match}$, and 456 (371) facts of *Movie* (*Book*) entities for the data fusion step $KnowMore_{class}$. The experimental evaluation in Section 7 is based on the averaged P/R/F1 scores from a 10-fold cross validation (90% training and 10% testing). To evaluate how performance is affected by the scale of the training data, we have conducted our experiments with subsets of the data which vary in size. In particular, for each type, we run 10 experiments where the subset of the training data uses *n* percent of the original training data set and *n* is in the range of [10, 100]. 10-fold cross-validation is performed for each *n*.

Figure 4 presents the results for $KnowMore_{match}$, where the X-axis indicates the size of the training data and the Y-axis shows the P/R/F1 scores. The F1 score reaches 0.8 (0.88) at 60% (40%) percent of training data, and the P, R and F1 curves become steady with training data sets of the size of 80%(50%) for the *Movie* (*Book*) type, i.e. training data sets of at least 130 (119) entity descriptions for the *Movie* (*Book*) type.

Similar characteristics can be observed for the $KnowMore_{class}$ approach in Figure 5, where the F1 score reaches 0.89 (0.87) at 70% (70%) of training data for *Movie (Book)*. Hence, results suggest that even with comparably limited amounts of training data, reasonable performance can be achieved, thereby supporting the application across types and datasets. For instance, the cost for retrieving 80% (70%) of our entity matching (data fusion) ground truth from CrowdFlower with the approach as described in Section 6.2, is less than 15 USD and the time required is less than 24 hours for each type.

8.2. Model Performance across Types

In this section, we assess the performance of *KnowMore* across different types, i.e. without a type-specific training phase. Thus, we merge the aforementioned type-specific datasets *Book* and *Movie* and perform a 10-fold cross-validation using the query sets for both types. The averaged performance of $KnowMore_{match}$ on this cross type dataset is P = 0.782, R = 0.892, F1 = 0.833, where the precision is lower than the type-specific results (Section 7), but the overall performance and F1 score is still comparable, the latter being slightly above the F1 score of the type-specifically trained *Movie* model (0.83). The result of $KnowMore_{class}$ is P = 0.902, R = 0.825, F1 = 0.862. Where the precision is higher than the type-specific result for *Book* model (0.886) and lower than the one from the *Movie* model (0.967). This suggests that our approach can work on models trained on cross-type data. Further studies are required with more diverse query as well as datasets, involving larger amounts of types, to fully validate this finding.

9. Discussion & Limitations

9.1. Potential of KBA from Web Markup

Beside the specific KBA task evaluated in this paper, where we aim at populating a fixed set of properties from a given KB schema, markup data shows significant potential to augment KBs with properties not yet present at all in KBs. Investigating the data from our two datasets (Movie, Book) and another set of 30 randomly selected entities of type *Product*, shows that a large proportion of statements involve properties not yet present in any of the KB schemas. For instance, for movies (books), 62.5% (66.8%) of entity descriptions in *F* contain facts not yet present in our set of mapped predicates. Comparing product descriptions from *F*, we detect 20.6% statements containing properties not yet present in the DBpedia ontology at all (verified through manual inspection).

In addition, we observe a considerable potential for augmentation of KBs with new entities, as opposed to augmenting existing ones. To assess performance in such cases, we randomly select 30 names of products under the requirement that each appears in at least 20 different PLDs in WDC, to ensure that there is sufficient consensus on the name being a legitimate product title. Manual inspection confirmed that none of such randomly selected products is represented in DBpedia. Table 10 shows the performance of $KnowMore_{class}$ and our baselines on this dataset.

Table 10
Data fusion performance for *Product* entities.

Approach	**P**	**R**	**F1**
KnowMore$_{\text{class}}$	**0.983**	**0.927**	**0.954**
PrecRecCorr	0.827	0.485	0.611
CBFS	0.876	0.686	0.769

Results indicate that the performance gain of our approach is particularly evident on such long-tail entities as represented in our *Product* dataset.

9.2. Limitations

Results demonstrate that *KnowMore* is able to exploit Web markup data for KBA tasks. Further improvement can be gained by applying our approach on a focused crawl, targeted towards a specific KBA task, such as movie enrichment, rather than a cross-domain Web crawl such as the WDC/Common Crawl.

In contrast to related KBA approaches such as [13] or [36], it is worth noting that our approach is trained for particular entity types only, not towards particular properties, as is the case with the aforementioned approaches. Hence, *KnowMore* can be adapted to a wider range of scenarios with less effort than previous KBA approaches. In addition, we have demonstrated in Section 8.2 that our models can potentially generalise across types.

Performance strongly differs between query sets, and hence, type-specific markup datasets what presumably is caused by the variance in quality and quantity of facts in the WDC corpus between distinct types. Particular challenges arise from entities with a large amount of co-references, where data usually originates from a wide variety of sources with varying degrees of quality. Compared to the baselines, our results indicate a particular strong performance gain of our approach in such cases.

One limitation is our exclusive focus on *schema.org* statements. This constraint is motivated by the costliness of providing high-quality schema mappings between markup statements and three KBs and the fact that *schema.org* is the vocabulary of most widespread use [19]. While *schema.org* adopters usually are motivated by the goal to improve their search result rankings, one assumption is that other vocabularies might show a different distribution of types and predicates, due to distinct motivations. This deserves deeper investigation as part of future work.

It is also worth noting that our KBA task setup ignored a large part of the markup data, i.e. 49.3% of facts in our type-specific subsets do not involve any of our selected *schema.org* properties. To consider other vocabularies, we are currently aiming at including a preliminary schema matching step with the intention of improving recall further.

Another important aspect concerns the temporal nature of fact correctness, specifically for highly dynamic predicates, such as the price tag of a particular product. While we do not consider temporal features as such, we argue that the dynamic nature of markup annotations is well-suited to augment particularly dynamic statements. This suggests particular opportunities for updating or complementing KBs with dynamic knowledge sourced from Web markup.

10. Conlusions and Future Work

We have introduced *KnowMore*, an approach towards knowledge base augmentation from large-scale Web markup data, based on a combination of entity matching, data fusion and diversification techniques. We apply our method to the WDC2015 corpus as largest publicly available Web markup crawl (approx. 20 billion quads) and augment three established knowledge bases, namely Wikidata, Freebase and DBpedia. Evaluation results suggest superior performance of our approach with respect to diversity as well as correctness compared to state-of-the-art data fusion baselines, with an F1 score increase of 11.7% respectively 6.5% compared to the two baselines across datasets. Our experimental results indicate comparably consistent performance across a variety of types, whereas the performance of baseline methods tends to vary strongly.

Our evaluation of the KBA task on two types demonstrates a strong potential to complement traditional knowledge bases through data sourced from Web markup. We achieve a 100% coverage for particular properties, while providing significant contributions to others. In addition, we demonstrate the capability to augment KBs with additional entity descriptions, particularly about long-tail entities, where for randomly selected entities of type *Product* from WDC, we are able to generate new entity descriptions with an average size of 6.45 facts.

While our experiments have exploited the WDC corpus, we will consider more targeted Web crawls, which are better suited to augment entities (or properties) of a particular type or discipline. Here, targeted datasets which are retrieved with the dedicated aim to suit a particular KBA task are thought to further improve the KBA performance. Another identified direction for future research is the investigation of the complementary nature of other sources of entity-centric Web data, for instance, data sourced from Web tables, when attempting to augment KBs.

Additional objectives for future work have surfaced during the experiments. For instance, identity resolution problems might occur during the matching step originating from different meanings of a particular entity. Current work aims at pre-clustering result sets into distinct entity meanings, from which we will be able to augment distinct disambiguated entity descriptions. Finally, we are investigating an iterative approach which enables the generation of entity-centric knowledge graphs of a certain length (*hop-size*), rather than flat entity descriptions. This would further facilitate research into the generation of domain or type-specific knowledge graphs from distributed Web markup.

11. Acknowledgments

This work has been partially funded by the European Union's Horizon 2020 research and innovation program (Grant No. 687916 - AFEL project). The authors would like to thank Robert Meusel for his inspiring feedback and support with the presented work.

References

[1] A. P. Aprosio, C. Giuliano, and A. Lavelli. Extending the coverage of dbpedia properties using distant supervision over wikipedia. In *NLP-DBPEDIA*, 2013.

[2] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys (CSUR)*, 41(1):1, 2009.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD/PODS*, 2008.

[4] V. Bryl and C. Bizer. Learning conflict resolution strategies for cross-language wikipedia data fusion. In *WWW*, 2014.

[5] L. Bühmann and J. Lehmann. Universal owl axiom enrichment for large knowledge bases. In *EKAW*. Springer, 2012.

[6] L. Bühmann and J. Lehmann. Pattern based knowledge base enrichment. In *ISWC*, 2013.

[7] 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 *SIGKDD*, 2014.

[8] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From data fusion to knowledge fusion. *VLDB*, 2014.

[9] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. Knowledge-based trust: Estimating the trustworthiness of web sources. *Proceedings of the VLDB Endowment*, 8(9):938–949, 2015.

[10] A. Dutta, C. Meilicke, and H. Stuckenschmidt. Enriching structured knowledge with open information. In *International Conference on World Wide Web*, pages 267–277, 2015.

[11] D. Gerber, S. Hellmann, L. Bühmann, T. Soru, R. Usbeck, and A.-C. N. Ngomo. Real-time rdf extraction from unstructured data streams. In *ISWC*, 2013.

[12] H. Ji and R. Grishman. Knowledge base population: Successful approaches and challenges. In *ACL*, 2011.

[13] P. H. Kanani and A. K. McCallum. Selecting actions for resource-bounded information extraction using reinforcement learning. In *WSDM*, 2012.

[14] T. Knap, J. Michelfeit, J. Daniel, P. Jerman, D. Rychnovskỳ, T. Soukup, and M. Nečaskỳ. Odcleanstore: a framework for managing and providing integrated linked data on the web. In *WISE*. Springer, 2012.

[15] J. Lehmann, S. Auer, L. Bühmann, and S. Tramp. Class expression learning for ontology engineering. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1):71–81, 2011.

[16] P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: linked data quality assessment and fusion. In *EDBT/ICDT Workshops*. ACM, 2012.

[17] R. Meusel and H. Paulheim. Heuristics for fixing common errors in deployed schema.org microdata. In *ESWC*, 2015.

[18] R. Meusel, P. Petrovski, and C. Bizer. The webdatacommons microdata, rdfa and microformat dataset series. In *ISWC*. 2014.

[19] R. Meusel, D. Ritze, and H. Paulheim. Towards more accurate statistical profiling of deployed schema.org microdata. In *ACM Journal of Data and Information Quality*, volume 8, 2016.

[20] J. Michelfeit and T. Knap. Linked data fusion in odcleanstore. In *ISWC*, 2012.

[21] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL & AFNLP*, 2009.

[22] H. Paulheim and S. P. Ponzetto. Extending dbpedia with wikipedia list pages. In *NLP-DBPEDIA*, 2013.

[23] D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, 2000.

[24] R. Pochampally, A. Das Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing data with correlations. In *SIGMOD*. ACM, 2014.

[25] P. Ristoski and P. Mika. Enriching product ads with metadata from html annotations. In *ISWC*, 2016.

[26] D. Ritze, O. Lehmberg, and C. Bizer. Matching html tables to dbpedia. In *WIMS*, 2015.

[27] D. Ritze, O. Lehmberg, Y. Oulabi, and C. Bizer. Profiling the potential of web tables for augmenting cross-domain knowledge bases. In *WWW*, 2016.

[28] P. Sahoo, U. Gadiraju, R. Yu, S. Saha, and S. Dietze. Analysing structured scholarly data embedded in web pages. In *WWW Companion*, 2016.

[29] A. Schultz, A. Matteini, R. Isele, C. Bizer, and C. Becker. Ldif-linked data integration framework. In *COLD*, 2011.

[30] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, 2013.

[31] A. L. Strauss. *Qualitative analysis for social scientists*. Cambridge University Press, 1987.

[32] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW*, 2007.

[33] D. Taibi and S. Dietze. Towards embedded markup of learning resources on the web: a quantitative analysis of lrmi terms usage. In *WWW Companion*, 2016.

[34] A. Tonon, V. Felder, D. E. Difallah, and P. Cudré-Mauroux. Voldemortkg: Mapping schema. org and web entities to linked open data. In *ISWC*, 2016.

[35] G. Weikum and M. Theobald. From information to knowledge: harvesting entities and relationships from web sources. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 65–76. ACM, 2010.

[36] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin. Knowledge base completion via search-based question answering. In *WWW*, 2014.

[37] R. Yu, B. Fetahu, U. Gadiraju, and S. Dietze. A survey on challenges in web markup data for entity retrieval. In *ISWC*, 2016.

[38] R. Yu, U. Gadiraju, X. Zhu, B. Fetahu, and S. Dietze. Entity summarisation on structured web markup. In *ESWC: Satellite Events*, 2016.

[39] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han. A bayesian approach to discovering truth from conflicting sources for data integration. *VLDB*, 2012.