Semantic Web 1 (0) 1–5
IOS Press

FarsBase: The Persian Knowledge Graph

Majid Asgari [a], Ali Hadian [a] and Behrouz Minaei-Bidgoli [a,*]
[a] *Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran*

Abstract. Over the last decade, extensive research has been done on automatic construction of knowledge graphs from Web resources, resulting in a number of large-scale knowledge graphs such as YAGO, DBpedia, BabelNet, and Wikidata. Despite that some of these knowledge graphs are multilingual, they contain few or no linked data in Persian, and do not support tools for extracting knowledge from Persian information sources. FarsBase is the first Persian multi-source knowledge graph, which is specifically designed for semantic search engines to support Persian knowledge. FarsBase uses a diverse set of hybrid and flexible techniques to extract and integrate knowledge from various sources, such as Wikipedia, Web tables and unstructured texts. It also supports entity linking, which allows integration with other knowledge graphs. To maintain a high accuracy for triples, we adopt a low-cost mechanism for verifying candidate knowledge by human experts, where the candidates for human verification are prioritized using different heuristics. FarsBase is being used as the semantic-search system of a Persian search engine and efficiently answers hundreds of semantic queries per second.

Keywords: Semantic Web, Linked Date, Persian, Knowledge Graph

1. Introduction

Construction of knowledge graphs (KGs) from open-access data such as Wikipedia has revolutionized the semantic capabilities of information retrieval systems, including search engines and personal assistants like Siri, Google Assistant, Alexa, and Cortana. Users mostly prefer to find *exact answer* instead of scrolling down through a list of results and then finding the answer in a Web page. For example, the desired response for “How many children does the Queen have?” is simply "Four". Such a response requires a credible and up-to-date knowledge graph with comprehensive information for answering semantic user queries. In fact, most of the challenges in information acquisition that were traditionally handled by the search engine’s users — such as credibility analysis of the information sources (“should I trust this website?"), conflict resolution, and fact-checking– should now be handled by knowledge graph systems.

The past decade have witnessed ambitious research in knowledge graph construction. This includes knowledge graphs constructed from Wikipedia such as DBpedia [1]; systems that extract knowledge raw text, e.g. NELL [2]; as well as the hybrid systems that exploit multiple types of information sources, including Yago [3].

In this paper we present FarsBase, a Persian knowledge graph constructed from various information sources, including Wikipedia, Web tables and raw text. FarsBase is specifically designed to fit the requirements of structural query answering in Persian search engines. Our contribution are as follows:

- We provide a hybrid architecture for knowledge graph construction from multiple sources that leverages both top-down and bottom-up approaches: A preliminary version of the knowledge graph is constructed from Wikipedia infoboxes, which is consequently used to extract more knowledge from other knowledge graphs, raw text, and tables.
- Contrary to other knowledge graphs, FarsBase is specifically constructed for Persian search engines. Therefore, the entire process of data collection, data filtering, and query processing is specifically designed to boost the user experience. In that respect, the query log plays a key role in prioritizing data sources, entities, classes, infoboxes, properties, and images; in different stages of the system. The workload-driven design of FarsBase

*Corresponding author. E-mail: b_minaei@iust.ac.ir

requires fewer experts for building a knowledge graph, because the human resources and system-tuning efforts focus on data records that are more important to the user, e.g. frequently searched.
- FarsBase supports rule-based methods that enable flexibility for data extraction and manipulation in several components of our architecture, including infobox extraction, raw text extraction, data transformation, and data cleansing.
- FarsBase supports efficient human labeling for managing and cleansing data from different sources and in multiple versions. It benefits from various types of metadata provided by the different extractors, e.g. the time-flags and the accuracy/confidence of different extraction modules for each triple. Such features can be used for prioritizing and grouping the entities for cost-effective batch verification of triples by human experts.
- We provide a mechanism for integrating data from heterogenous knowledge extractors. Our mechanism handles different versions from data sources with minimum expert intervention. This requires extracting temporal facts and triple versioning for handling further conflicts between the new and current information. To the best of our knowledge, FarsBase is the only multi-source knowledge graph that supports timeliness[4] by handling different versions of data from multiple sources.

The remainder of this paper is organized as follows. The preliminaries and motivation is briefly introduced in section 2. Section 3 describes a cost-based solution to select knowledge sources for FarsBase. We give an overview about FarsBase architecture in section 4. Section 5 explicates knowledge extraction from different sources, including Wikipedia, Web table and raw text. In sections 6,7, we describe how extracted triple are mapped and integrated into a unified knowledge graph. Evaluation and statistics about FarsBase are reported in Section 8. Section 9 describes related work in knowledge graph construction, quality assessment, mapping, relation extraction from raw texts, never ending paradigms and knowledge augmentation. Finally, section 10 concludes the paper with directions for future work.

2. Preliminaries and Motivation

In this section, we briefly introduce the basics of knowledge graph construction and representation. Also, we explain challenges for constructing a multi-domain Persian knowledge graph.

2.1. Knowledge Base and Knowledge Graph

A knowledge base contains a set of facts, assumptions and rules that allows storing knowledge in a computer system. Knowledge bases can be specific to certain domains, e.g. a medical knowledge base containing facts about medical drugs (such as their properties and interactions). Also, knowledge from multiple domains can be integrated to build a general-domain knowledge base. For example, DBpedia [1] is a multi-domain knowledge base that is semi-automatically constructed from Wikipedia articles. Knowledge bases require a data model to organize the facts. A typical approach is to define an ontology, where data instances (a.k.a. *entities*) are assigned to classes. Each class can be a subclass of another class, which results in a hierarchy known as ontology tree. The facts of a knowledge base are commonly represented using a knowledge representation format. Modern multi-domain knowledge bases use the Resource Description Framework (RDF) for knowledge representation. RDF is primarily designed to represent resources on the Web, but it can also be used for knowledge management and supports essential features for constructing a knowledge base, such as Is-A relations and object properties.

In Semantic Web and linked data, there are different definition of *knowledge graph* (KG); Ehrlinger et al. tried to clarify the term in [5]. They mentioned 5 selected definitions of knowledge graph and presented an architecture for it. They assumed a knowledge graph is somehow superior and more complex than a knowledge base because it contains a reasoning engine and also integrates knowledge from one or more sources.

2.2. Resource Description Framework

RDF is a standard for conceptualizing structural data. In this model, data is represented as a set of triples consisting of a *subject*, a *predicate*, and an *object*. A set of triples forms an *RDF graph*.

The RDF format enables knowledge representation using Web resources, where each resource has a Unique Resource Identifier (URI). In RDF, subjects and predicates are URIs, and objects can be either URIs or literal values. RDF data is serialized and stored using different textual syntaxes, e.g. Turtle and NTriples.

For example, the fact that “Einstein knows Niels Bohr” can be represented in Turtle syntax as follows :

```
<http://example.name/Albert_Einstein>
<http://xmlns.com/foaf/0.1/knows>
<http://example.name/Niels_Bohr> .
```

RDF can be easily used for knowledge graphs derived from non-English data. String literals can have a language tag, which is very useful for building multilingual knowledge graphs. For example, Albert Einstein can be represented as "Albert_Einstein"@en or "آلبرت_آنیشتین"@fa.

2.3. FarsBase: A Persian Knowledge Graph

Constructing a multi-domain and comprehensive knowledge graph from unstructured and semi-structured Web contents has been of interest for a while. The DBpedia project was initiated a decade ago to construct a knowledge graph from Wikipedia. Further works like Yago[3] and BabelNet[6] integrated other sources, e.g. WordNet and GeoNames[7], in order to construct a more-enriched knowledge graph. Raw text data collected from Web can also be used to supplement knowledge graphs [8].

Thanks to tons of research in knowledge graph construction techniques, multi-domain knowledge graphs such as DBpedia and Yago have a comprehensive set of facts extracted from English and European languages. However, these knowledge graphs do not contain enough facts from a low-resource and challenging language[9], such as Persian. This is mostly due to the fact that tools required for automatic knowledge graph construction are not mature enough to be used in multilingual knowledge extraction engines. Also, Persian NLP toolsets suffer lower accuracies due to the small size of the Persian corpora. We tackled these challenges by various techniques that boost the accuracy of knowledge graph construction in Persian. The main challenges for FarsBase construction are summarized as followed:

- Knowledge graph construction engines adopt state-of-the-art methods for extracting knowledge from raw text and semi-structured data such as Web tables. Essential software libraries include entity linking, base phrase chunking, dependency parsing, coreference resolution, and entity disambiguation. The errors caused by each of the tools will be propagated throughout the rest of the system and results in low precision results. Whenever the accuracy of standard NLP toolsets do not suffice for certain subtasks in knowledge extraction, the inaccurate methods should be either enhanced or replaced with an alternative approaches to deliver enough accuracy for knowledge graph construction.
- Knowledge graph construction requires human supervision as a part of the process, e.g. in the mapping phase (explained later in section 6). DBpedia and other projects mostly focused on English and other widespread languages, and did not put effort on mapping and cleaning Persian data. We argue that human supervision for yet-another language is non-trivial. For example, the human supervision process for dependency parsing (used for knowledge extraction from raw text) is entirely different in Persian.
- Having an ideal set of high-precision NLP toolkits, it is challenging to extract *enough* knowledge from Persian Web sources that satisfies a certain application. In particular, FarsBase is primarily constructed to be used as a backbone for semantic search in Persian Search Engines. This requires that the knowledge graph be accurate for user queries, specifically for the *frequent* queries. Also, since a significant number of user queries target *recent* knowledge (e.g. details about a new celebrity or a recent event), the knowledge construction mechanism should be specifically eager to extract knowledge from the trending entities and relationships. To the best of our knowledge, none of the accessible knowledge graphs are optimized with respect to user search query log.

FarsBase is designed to be specifically precise on parts of the knowledge graph that are *frequently searched*. Aside from having precise knowledge for frequent queries, a significant share of structural queries in a search engine correspond to *recent* content, such as queries about new celebrities and recent events. Therefore, our knowledge construction engine extracts new entities and relationships that are recently introduced in Persian Web sources.

FarsBase is automatically constructed from Persian (Farsi) section of Wikipedia, and is expanded by other data sources such as text and Web tables. Despite the fact that FarsBase is connected to various multi-lingual knowledge graphs, the main focus of FarsBase is on extracting knowledge from Persian sources. In fact, some multilingual information extraction tools already support Persian, but their accuracy on Persian sources is very low. In DBpedia, for example, the knowledge extraction engine is entirely run on Persian Wikipedia

but very few triples are extracted. However, FarsBase is primarily constructed to extract Knowledge from Persian sources on the Web.

FarsBase has been designed to be a multi-source knowledge graph. Even though other knowledge graphs also use multiple sources, none of them is designed to be a knowledge graph which accepts structural data and raw texts concurrently. For example DBpedia mainly extracts triples from Wikipedia (Some researches has been made to augment DBpedia from other sources e.g. [10],[11],[12]). Similarly, BabelNet is constructed with Wikipedia and WordNet only; and NELL has mainly focused on raw texts and extracts limited number of predicates. To our knowledge, Yago is the only knowledge graph that supports multiple structural sources and can extract from raw text using an extension [13]. However, the libraries and architecture of Yago requires language-specific libraries that are of very poor quality for Persian.

3. Knowledge Sources

FarsBase is constructed by integrating data from different sources, including Wikipedia infoboxes, Tables extracted from Web pages, and raw text collected from various sources.

The information sources for building a knowledge graph should be rich and accurate. Different types of input sources may be considered for knowledge graph construction, including:

- Existing knowledge graphs: Freebase, Wikidata.
- Encyclopedias: Wikipedia, World Book Encyclopedia
- Domain-specific databases: IMDb, BrainyQuote, TripAdvisor, etc.
- Semi-structured Web sources: Web tables, infoboxes.
- Unstructured (raw) text, collected from the Web and book texts.
- Direct human input: collected by human experts or crowdsourcing.

Each of the existing knowledge graphs used one or a couple of the above sources that were accessible at the time of knowledge graph construction. Table 1, summarizes the input sources for common knowledge graphs.

3.1. Availability of Sources in Persian

Sources for constructing Persian knowledge graphs are limited, both in terms of the number of available options, and their quantity and quality. Wikipedia, for example, contains over 5.6 million English articles, but only 0.6 million articles in Persian. Aside from Wikipedia and raw text, the other common data sources for knowledge graph construction, shown in Table 1, have no Persian alternative or are so small that do not worth extraction. For example, there are few Persian websites similar to the IMDB movie database, such as `SourehCinema.com`, but their databases is almost a subset of the information already available in Persian Wikipedia.

Persian sources are sparse. In Wikipedia, for example, English articles usually contains more information than their Persian version. More specifically, Persian Wikipedia articles are less verbose, have fewer links and many of them have no infoboxes. This is mostly due to the smaller community of contributors for Persian, which also impacts the quality of the content. Other types of sources such as Web pages are even poorer in quality due to the massive amount of hoaxes and false information, that even appear on credible news sources.

Due to the numerous challenges associated with Persian sources, the first step for FarsBase construction is to select a set of sources that have comprehensive and accurate information.

3.2. Source Selection: A Cost-based Approach

In order to use FarsBase for query answering in search engines, its primary application, it should contain a diverse and comprehensive set of facts from all domains to cover a large share of user queries. Such a large knowledge graph should be collected with an affordable effort, hence we should consider the cost of knowledge extraction from each source. Extracting information from texts, tables, and other sources on the Web is far more challenging and costly than the structured sources like Wikipedia, mainly due to the lack of structure and quality issues.

General cost measures: The *cost* of exploiting a source can be defined as the human effort required for verifying its facts, e.g. in terms of "*second per tuple*". Using an incredible data source might lead to false information and increase the demand for human verification. Errors generated by the knowledge extraction modules can also call for more verification.

Table 1
Sources used in famous knowledge graphs

Knowledge Graph	Data Sources
Knowledge Vault	Freebase, Wikidata, Wikipedia, Raw text, Tables and Web pages, Human knowledge
Wikidata	Wikipedia, Human knowledge
Freebase	Wikipedia, Domain-specific Databases like NNDB, Fashion Model Directory, MusicBraiz, ...
DBpedia	Wikipedia
BabelNet	Wikipedia, WordNet
Yago	Wikipedia, WordNet, GeoNames
NELL	Raw text, Crowdsourcing
Google Knowledge Graph	Wikidata, Freebase, Wikipedia, CIA World Factbook, User feed-backs
Microsoft Knowledge Graph	Freebase, Wikipedia, LinkedIn, User feed-backs, Online Databases like Foursquare, TripAdvisor, Yelp, BrainQuote, IMDB, ...

Application-specific cost measures. In most applications, facts of the knowledge graph are not of the same importance. In a search engine, for example, the importance of an entity or relationship is proportional to how *frequently* it appears in user queries. A false relationship about frequently-searched entities, e.g. "Barack Obama was born in Africa", has a higher impact on users than the same false information on non-famous people. Moreover, the cost of extracting a false relationship is much higher than missing a relationships from a source, which should be considered when tuning the sensitivity of the extraction algorithms.

We considered the following criteria for selecting and prioritizing the sources

1. **Quality of source**: Some sources such as blogs and many news websites contain a high share of hoaxes and wrong statements. The amount of false information in a source should be quite low, say <5%, so that the data can be verified in batch or using automated methods.
2. **Precision of the available extractor**: Information extraction from raw text and Web tables is more challenging compared to structured sources such as databases and Wiki infoboxes. Only few knowledge graphs are constructed based on raw-text, most notably KnowledgeVault and NELL. Such knowledge graphs either use crowdsourcing (which needs a high amount of user contribution), or heavily rely on other knowledge graphs, e.g. Freebase and Wikidata, which are not available for Persian. Moreover, the difficulty of the extraction task depends on the maturity of state-of-the-art NLP tools. Persian NLP libraries cannot efficiently extract knowledge, especially if the content has a high rate of misspellings and slang words. The immaturity of Persian NLP tools, along with the low quality of the Persian sources, makes it very challenging to extract high-quality knowledge in Persian.
3. **Verification cost**: Knowledge extraction should ideally be an automatic process. If the precision of the extracted information is higher than a certain threshold, say 95%, it could be permissible to accept all extracted data without manual verification. For lower qualities, however, the extracted information should be verified by human experts or human-in-the-loop labeling methods [14, 15].
4. **Overlapping of sources**: If a relationship is available in multiple sources, we prefer to extract it from the source with the lowest cost. Considering that most of the knowledge in FarsBase comes from encyclopedias such as Wikipedia, for other sources we are interested in how much *extra* information can the new source add to the existing knowledge graph.

To achieve a good trade off between the quality and quantity of the extracted knowledge, in the following we provide a brief cost analysis for each of the available sources for Persian knowledge extraction.

3.2.1. Wikipdia

Wikipedia is a rich knowledge resource developed by millions of contributors. Most of the multi-domain knowledge graphs such as DBpedia, Yago, and BabelNet use Wikipedia as their primary knowledge source [4]. Although the Persian Wikipedia is smaller than the English version, it is still the most valuable and accurate Persian knowledge source in comparison with other sources. Many Wikipedia articles have one or more infoboxes. Knowledge graph construction from infoboxes is very cost-efficient, because the

human effort is mostly on the mapping and transformation (sec 6.1) and the extracted data are accurate enough such that no human verification is required on the extracted triples.

Aside from the structured data in Wikipedia (infoboxes, abstracts, categories, redirects, etc), the raw content of the articles is very rich and has the highest quality compared to other raw text sources. This is further discussed in section 3.2.3.

3.2.2. Web Tables

Web tables are also rich sources for batch triple extraction. One can easily extract a considerable number of entities and their relations. Due to the high variety of structures in Web tables, the cost of extracting information from tables is much higher than infoboxes because it need entity linking [16] and special extraction approaches[17–19].

3.2.3. Raw Text

Due to the limited size of Persian Wikipedia, it has to be supplemented with additional knowledge extracted from other sources, most notably the raw text and Web tables. Hence, it is very important that the knowledge source contains *substantial amount of information* with *high quality* and *low complexity*, e.g. not having complicated grammar or table structure, so that the available tools can extract the knowledge effectively.

Raw-text extraction can be applied on any type of raw text, including Web content, books, and OCR-extracted texts. However, raw text requires significant effort to extract and verify knowledge because each triple must be verified by human experts. To overcome the high cost of extraction, we should select the most informative texts that contain a high number of entities and relationships.

To select the most cost-effective sources as input for RTE, we investigated four types of raw-text data: Wikipedia article bodies, news articles, technical and personal blog posts. We picked a set of randomly selected articles from each sources, and manually counted the number of triples that are suitable for the knowledge graph. The data from all sources is collected on November 1, 2016.

To investigate the *difficulty* of knowledge extraction from each raw text source, we investigate the impact of co-reference resolution for each data source. Co-reference resolution can help extractor to find out more triples from the raw texts, but the quality of state-of-the-art Persian NLP toolsets for co-reference resolution is not very high. In all of experiments, experts have counted number of triples with and without co-reference resolution.

Table 2
Number of triples per 1000 words for different raw text sources

Source	# Triples	# Triples with C.R.
Wikipedia	46	68
News Sites	3	3.3
Technical blogs	1.4	1.6
Literary blogs	0.2	0.26

We performed a few preprocessing steps on each data source to ensure a valid comparison:

- **Wikipedia article bodies:** We randomly sampled 50 Wikipedia articles with a minimum length of 500 words, and triples were manually extracted by experts.
- **News articles:** We sampled 50 news articles from a famous news agency website (تابناک)[1]. Note that most Persian news websites contain a large amount of auto-generated contents, such as tables, summaries of stock prices, weather forecasts, currency exchange rates, gold prices, etc. Since the auto-generated values reflect daily events and are not valuable for KGs, we excluded them from the body of the samples. We observed that most news articles contain a very small number of useful triples.
- **Blogs:** We sampled 50 posts from Persian technical weblogs, and another 50 samples from literary weblogs.

3.2.4. Evaluation of Raw Text Sources

Table 2 shows the average number of triples in each category of the raw text sources, measured by human experts with and without co-reference resolution. Results are normalized by length to represent the number of triples per 1000 words.

Table 3 summarizes different cost aspects of the available data sources for FarsBase, including the need for natural language processing (especially Co-reference resolution), number of triples, quality of each triple, and the overall cost of extracting knowledge from the source. Considering all these parameters, we built FarsBase using structured parts of Persian Wikipedia (primarily infoboxes); along with a selection of Persian news websites to cover recent information. We also support a high-precision approach to extract data from Web tables, which requires explicit human supervision.

[1] http://tabnak.com

Table 3
Cost of available sources for FarsBase

Source	Co-references	Number of Triples	Accuracy of Triples	Cost	Used in FarsBase
Wikipedia (excl. raw text)	-	Very High	Very High	Affordable	Yes
Web-Tables	-	Very High	Very High	Very High	Limited
News sites (text)	Average	Low	High	Very High	Yes
Formal/Technical Weblogs (text)	Low	Low	Low	Very High	No
Personal Weblogs (text)	Low	Very Low	Very Low	Very High	No

In more details, FarsBase is constructed from the following data sources: Wikipedia structured data (infoboxes, entity categories, ambiguities, external links, Web pages, images, connections between entities, and rewriting ids, etc.), tables (with explicit human labeling), Wikipedia article bodies, and news websites.

4. Architecture

In this section, we present the FarsBase system architecture and illustrate how integrating the knowledge graph with a search engine — especially having access to a query log— can optimize both the knowledge construction and the knowledge-graph search systems.

FarsBase consists of a KG construction system for extracting knowledge, as well as a search system for answering KG-based user queries and suggesting similar entities in a search engine. Figure 1 shows an overview of the FarsBase architecture.

The construction and search systems benefit from shared components and data in a symbiotic manner: The construction system updates the knowledge graph with the most recent triples, and this helps the Resource Extractor to extract more entities and ontology predicates from the input sources to feed the construction system.

4.1. Knowledge Graph Construction

FarsBase extracts knowledge from various types of sources, including Wikipedia, semi-structured Web data (e.g. Web-tables) and raw texts. The extracted data is then cleaned and organized with partial supervision by the expert.

Extractors. Each resource type requires a separate extractor to process its data format. The main three extractors are as follows:

- Wikipedia Extractor (WKE): Extracts triples from Wikipedia dump, including the infoboxes, abstracts, categories, redirects, and ambiguities.
- TaBle Extractor (TBE): Extracts from Web pages that contain tables with informational records. The schema of the table, i.e. the mapping to KG, is suggested by the system but needs to be verified by a human expert. The current version of TBE is mostly designed for parsing tables in Wikipedia, where the mapping is easier to infer.
- Raw-Text Extractor (RTE) consists of four extractor modules, namely the rule-based, distant supervision, dependency pattern and unsupervised modules. RTE continually extracts new triples for the KG. To prioritize the input sources, KG prioritizes input sources (websites) based on the facts that are under-demand. To provide raw-text data for RTE, some crawled sources by a search engine has been fed it. RTE consumes and archives fed raw texts.

Each extraction module preserves a variety of metadata, including the version and URL of the source that the triple is extracted from. For example, when WKE reads a new dump to generate triples, the version of the dump is stored along with all the extracted triples.

Source Selector. RTE is fed by the crawler of the search engine, although it can use an independent crawler such as Heritrix or Apache Nutch. The *Source Selector* module prioritizes Web pages based on the cost-efficiency of the sources, as discussed in section 3.2. The module also exploits the query-log of the search engine to prioritize Web pages based on the frequency that they appear in search results and the click-through rates.

Experts. The extracted predicates must be mapped to a unified ontology. The mapping table is incrementally constructed by the human experts and stored in Candidate Fact and Metadata Store (CFM-Store). The experts are also used for various verification tasks, most importantly for verifying the mapping schemata for TBE and all the triples extracted by RTE.

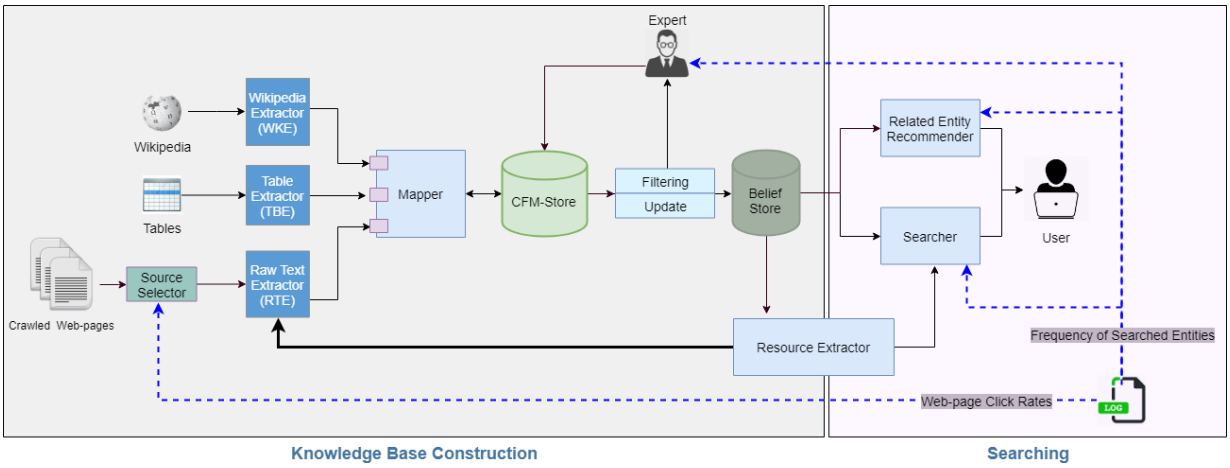

Fig. 1. FarsBase architecture

Mapper. The Mapper integrates triples generated by all extractors, and converts them into mapped triples. The mapped triples and their corresponding metadata are written to the CFM-Store. The metadata of triples includes document URLs, the module that extracted the triples, version information, date of extraction, source texts (if available), expert votes and notes.

Candidate Fact and Metadata Store (CFM-Store). CFM-Store stores all triples and their metadata. FarsBase supports versioning, i.e. the CFM-Store contains multiple versions of each triple. The latest version of triples, after expert approval, is updated/filtered periodically at specified intervals (currently once a day) to the Belief-Store without any metadata.

Filtering/Updating Triples. When transferring the triples to Belief-Store, the *Update Handler* re-builds the Belief-Store using the *latest version* of the mapped triples. Therefore, if a relation is removed from the input sources, it will no longer be written in Belief-Store.

Resource Extractor. *Resource Extractor* is used by both the extraction engine (specifically for RTE), and the search system. It extracts all possible entities and relationships from a given text in a brute-fore manner. Resource extractor does not apply any filtering or disambiguation on the extracted entities. Rather, it extracts all combinations of resources (i.e. entities and relationships) that might even overlap with each other, and further computes several features for each resource such as its position in the sentence and the class of the resource. These features are very helpful for filtering out irrelevant resources in the consecutive components, e.g. for disambiguation.

Note that the RTE relies on the resource extractor, which in turn is backed by the Belief-Store. Therefore, the Belief-Store should be initially filled with data from Wikipedia and Web-tables, and then the resource extractor can be used to run RTE.

4.2. The Search System

FarsBase is designed to respond to *unstructured queries* with structured output (entities) as the result. While the details of the search system is beyond the scope of this paper, we briefly explain the main components.

4.2.1. Searcher

Much of the attempt to search over knowledge graphs is on question answering systems, where the goal is to retrieve snippets of the documents that are related to the question and contain the answer [20]. However, we aim to design a system that answers specific questions with highly-reliable results instead of returning snippets.

An important requirement in a KB-based search in search engines is that *wrong results are intolerable*, such that a single wrong result by the KG-based search system for "President of the US" could be regarded as a disaster for the entire search business. As a result, the KG search system is admissible only if the precision is very close to 1. With this in mind, we follow a template-based approach which is similar to TBSL [21], where text queries are mapped to a structural SPARQL queries according to a set of templates. Unlike [21, 22], we do not directly allow auto-generated templates to be fed into the searcher. Instead,

we first generate a set of templates, and then each template is verified by human experts to ensure that the precision of FarsBase for the query is satisfactory. To handle the very large number of generated templates, we exploit the query log to rank templates based on how frequently they are triggered, and then the most frequent templates are verified by the experts.

4.2.2. Related Entity Recommender

Entity recommendation, i.e. suggesting related entities for a given entity, is an interesting service for search engines. FarsBase exploits relevance propagation through heterogeneous paths in the knowledge graph to estimate the relevance between entities. The training is done using an active learning approach.

5. Extraction

FarsBase extract knowledge from three types of data sources, namely Wikipedia, Web-tables, and raw text. We briefly explain each extractor, and the specific information that can be extracted from each source.

5.1. Wikipedia Extraction

The Wikipedia extractor in FarsBase is similar to DBpedia [23] and Yago [3]. The most important data types extracted from Wikipedia are as follows:

Infobox Data. Infoboxes contain a list of attribute and values. Note that an attribute might contain more than one value, e.g. “occupation: [job1, job2, ...]”. A value can be a string literal, an image, a link to other Wikipedia articles, an external link, or a combination of these types. The values can also be *objects* that contain a list of key-values.

Abstracts and Body of Articles. The body and abstracts of Wikipedia articles contain *internal links*, which are valuable sources for raw-text and table extractors. In particular, internal links are used to automatically extract high-confidence *patterns* for extracting facts from raw-texts.

Redirects. Entities may have multiple titles. For example, “سعدی شیرازی” (Saadi Shirazi, a Persian poet) has multiple alternative titles that refer to him in Persian Wikipedia, such as “استاد سخن” (The Master of Speech) and “افصح المتکلمین” (The Most Eloquent Speaker).

Disambiguation pages. Different entities might have a common title, such as people with the same name. For each ambiguous term, there is a disambiguation page that specifies all entities that might refer to it. Disambiguations and redirects of Wikipedia are specifically helpful for accurate entity resolution in other extractors.

Categories. Each Wikipedia article has one or more categories, e.g. "Saadi Shirazi" belongs to several categories such as "Persian poets of the 13th century" and "Persian Poets". Despite the fact that categories have some sort of hierarchy, e.g. might have subcategories, the categories are not well-structured and can be treated as a set of *tags* assigned by different people to each entity. Nevertheless, data analytic applications such as related-entity recommendation or KG search can leverage these labels. Moreover, various enhancements can be applied to improve the quality of category labels [24].

Images. The images of entities can be used for constructing the image-based FarsBase. The details of such system do not fall in the scope of this paper.

Inter-language and Inter-KG Links. Articles belonging to the same entity in different languages are linked to each other in Wikipedia. Also, other knowledge graphs and ontologies might have links to Wikipedia articles. Entity linking is further done in the mapping stage to integrate all records from different sources that belong to the same entity.

5.2. Tables

Information extraction from tables is a challenging task, specifically because the schema of Web tables is very flexible. Indeed, Web tables are primarily designed for viewing purposes, not to store a collection of data, such that many tables have split and merged cells. In most long tables, the subjects of the underlying triples are in a specific column of the table, albeit the subjects might also be in a specific row. More importantly, it is not trivial to recognize if a table contains significant information to be extracted. Despite these challenges, some research has been done to automatically extract entities or triples from tables [16, 19, 25].

5.3. Raw Text

The Raw Text Extractor (RTE) extract triples from unstructured text based on the current knowledge of

the KG. Even though FarsBase's RTE engine shares some aspects with never-ending learning (NELL), it is primarily designed to supplement the KB with information that are missing from Wikipedia. The triples supplied by the RTE should be almost of the same accuracy as the information extracted from Wikipedia, in order not to degrade the quality of the KG for semantic search. Therefore, RTE mostly adopts methods that maintain a high-precision, such as rule-based approaches, even though it can sacrifice the recall. Note that while all triples extracted by RTE are verified by the experts, evaluating highly-accurate results requires much less effort because the triples can be grouped and verified in batches.

Before extracting the triples from the raw text, some preprocessing steps must be applied on the text, including sentence boundary detection, word tokenization, part of speech (POS) tagging, base phrase chunking, dependency parsing, co-reference resolution and entity linking. Most of developed tools for these preprocesses in Persian language are not accurate enough. Even for the basic NLP tasks, such as NER, the available tools for Persian are not mature enough [26–28]. The errors caused by Persian NLP tools are propagated causes error propagation in triple extraction process, which can result in poor extraction quality.

While English RTE engines enjoy numerous tools for text preprocessing, FarsBase can only use the very few tools available for Persian, and in some cases, we developed our own tools specifically built for RTE. For example, there are no high-precision Co-reference resolution and entity linking modules for Persian, so we developed these libraries with only the required functionalities for RTE. Moreover, a base phrase chunker (BPC) could be very useful for relation extraction, but there is no reliable library for BPC in Persian, thus we did not use BPC for preprocessing the RTE input. For other preprocessing tasks, we used the JHazm[2] library.

FarsBase has four modules for raw-text triple extraction, namely rule-based extraction, distant supervision, dependency patterns, and unsupervised extraction. In the following, we briefly describe how different RTE methods pre-process and extract triples in FarsBase.

5.3.1. Entity Linking

Entity linking is the task of linking the entities mentioned in raw text to their corresponding KG URIs. It is a very essential task for triple extraction, because the subject of a triple must be linked to an entity URI. Once the URI of an entity in the text is known, we can leverage its type (ontology class) for further post-processing and filtering.

Entity linking in RTE uses the resource extractor module, which finds the links of all resources in a given input text. More specifically, the entity linker identifies the entities, categories and ontology predicates in the text, based on the known labels in the KG. Each resource may have more than one label, and a single word in the text may be shared between more than one resource.

The resource extractor does not perform any disambiguation analysis and merely finds the labels of all candidate resources. For example, a phrase may be linked to more than one entity, or even some verbs and adverbs may be mistakenly detected as entities. This is mainly because the modules that use the resource extractor, such as the search system and RTE, have different requirements and thresholds for disambiguation.

RTE has a separate module, named *entity linker*, which takes as input the extracted URIs and disambiguates the entities in the sentence, such that each word is linked to one and only one entity. The entity linker removes other type of resources including the predicate and category links. The entity linker uses the following heuristics for disambiguation :

Filtering by POS Tags. The POS tag of each entity is determined by the POS tagger. If a link contains just one word with certain POS tags, it will be eliminated. The set of POS tags consists of P, Pe, CONJ, POSTP, PUNC, DET, NUM, V, PRO and ADV. Labels of many entities in FarsBase are homographs with verbs or prepositions, e.g. "می رود" can refer to a village (می رود) or it can be the present continuous form of the verb "Go". Similarly, "به" can refer to a prepositions or a fruit (به).

Handling Homographs. Homographs are words that are written the same, but have different meanings. For example, even though the word "very" is a common modifier in English, in a few portions of texts it may refer to other entities such as the "Very" company. State-of-the-art NLP methods are still not very reliable for disambiguating such rare homographs [29]. Therefore, specific and rare entities are ignored using a manually-created list.

Class-specific Filters. Some entities have a very generic name that may cause a high level of ambiguity

[2] https://github.com/mojtaba-khallash/JHazm

for RTE. For instance, "چهل سالگی" (*"At the age of 40"*) is a famous Persian movie, although it can be a part of a sentence, e.g. "Alex died at the age of 40"). Such entities are very common in special classes, like *movies* (movie names), books, and artworks. To alleviate the ambiguity issue, if the detected entity belongs to certain classes, such as *Movies*, we will look for more evidence in the surrounding context using a reference list. For example, if a sentence contains "At the age of 40" (as a movie name), we require that the surrounding context also contains phrases such as "movie", "channel", "video", etc.; otherwise the linking will be ignored.

Context-based Disambiguation. This type of disambiguation is based on the context of the words. If a word is linked to more than one entity, its context is compared with the body of the corresponding Wikipedia articles of each entity. The cosine similarity between the context of the word and the body of the Wikipedia article is used as the measure to sort the entities.

Graph-based Disambiguation. This heuristic leverages the hyperlinks between the Wikipedia articles. If a word is linked to more than one entity, we consider the Wikipedia pages of the entities and score the ambiguous entities based on the number of hyperlinks between each entity and the Wikipedia pages of the surrounding words.

Combining Graph- and Context-based Disambiguations. We use the weighted sum approach to combine our graph- and context-based disambiguation heuristics. The score of each ambiguous entity is computed as $\alpha S_g + (1-\alpha)S_c$ where S_g and S_c are the similarity scores from graph- and context-based approaches, respectively. Note that the scores from each of the heuristics are normalized before combination. The parameter $\alpha = 0.3$ is tuned by trial and error. The entity with the highest similarity is linked to the word, and other entities are added to the word's "ambiguity-list". Also, if the scores of all ambigouos entities are lower than a pre-defined threshold, i.e. when the heuristics are too uncertain, none of the entity candidates are linked to the word, but all of them are added to the ambiguity-list.

The precision of FarsBase entity linker, evaluated by human experts on 30000 words, was 67.8%.

5.3.2. Pronoun Resolution

As shown in Table 2, co-reference resolution increases the recall of triple extraction up to 46%. To have a basic co-reference resolution, we used a baseline algorithm for pronoun resolution[30]. In our baseline, for each pronoun in the text, we choose the closest entity that is before the pronoun as the co-reference.

To detect the gender of person entities, we used a gazetteer (originally developed for a named entity recognizer) that contains a list of male and female first names in Persian, Arabic, Turkish, English and French languages [3].

5.3.3. Rule-based RTE

The rule-based RTE is a high-confidence extractor, which is based on Stanford TokensRegex [4][31]. A rule (i.e. pattern) is composed of words and entity classes. For example, the following rule suggests that if a sentence contains "پایتخت(capital)" followed by an entity of type "Country" (say E_{sub}), then followed by "شهر(city)" and another entity of type "Settlement" (say E_{obj}); then we can extract the relation (E_{sub}, fbo:capital, E_{obj}).

```
{word:/پایتخت/}]
(?\$subject [{ner:/.+Country.*/}])
[{word:/{شهر/}}]?
(?\$object [{ner:/.+Settlement.*/}]+)
```

5.3.4. Distant Supervision

Distant supervision is a semi-supervised approach for information extraction from raw-text [10, 32]. It is based on the intuition that the facts in the knowledge graph already have instances mentioned in the raw text sources. By aligning known facts, i.e. triples, with the corresponding sentences, we can automatically create a training set, so that a classifier can be trained to extract more facts with similar patterns from the raw text.

Inspired by Aprosio et al. [10], we developed a training dataset by automatically aligning FarsBase known facts with Persian Wikipedia's text. Out of 2.5 Million sentences from Persian Wikipedia articles, 172,368 sentences were mapped to the facts in FarsBase. The mapped sentences were evaluated by human experts, and 16,745 sentences were verified and added to the RTE engine. To improve the accuracy, we added a number of negative examples, i.e. sentences without any relationship between their entities.

We used LibSVM[5] to train a SVM classifier with bag-of-words features. The classifier uses the first K

[3] https://github.com/majidasgari/ParsNER
[4] https://nlp.stanford.edu/software/tokensregex.html
[5] https://www.csie.ntu.edu.tw/ cjlin/libsvm/,chang2011libsvm

words before/after the first/second entity (4 feature sets), as well as the POS tags between the two entities, and the classes of the entities.

The precision of the classifier was 54% on the gold data. To increase the precision, we used a type-checking algorithm, which eliminates triples that are not compliant with the domain and range constraints of their predicates. For example, the subject of a *nationalTeam* must be a *Person*, and its object must be a *Team*. Therefore, if the extracted rule is "Iran nationalTeam Africa", it will be eliminated due to incompliance with the domain constraints.

5.3.5. Dependency Patterns

Dependency patterns is a novel method for RTE which attempts to extract triples using "unique dependence trees". By definition, two dependency parse trees have the same dependency pattern if replacing each word with its corresponding POS tag, generates two identical trees. Figure 2 shows two sentences with the same dependency pattern.

Triple extraction with dependency patterns is based on the intuition that if a sentence contains a triple, other sentences with the same structure also contain similar triples. In such cases, the subject, object and predicate can be extracted from the words with the same position indexes in all sentences.

Note that the triples extracted by this method are not linked to the entities. Instead, a mapping module must link the subject and predicate to the KG resources.

RTE with dependency patterns is a supervised method, and human experts must define the position of the subject, predicate, and object in the sentences. Currently, 2000 most frequent dependency patterns are extracted automatically from Wikipedia texts and are annotated by the experts. Using these patterns, we extracted 240320 triples from Wikipedia articles.

5.3.6. Unsupervised RTE

We introduce an unsupervised method for triple extraction, which is based on dependency parsing and constituency tree. Unfortunately, there are no accurate libraries for constituency parsing in Persian. Our unsupervised method takes the main phrases of a sentence as input, and uses the dependency parse tree to detect the main phrases.

We explain our unsupervised RTE by an example. Consider the sentence shown in Figure 3. Given the dependency parse tree and the constituency tree, consider the following definitions:

- *Verb Phrase (VP)*: A phrase that contains at least one verb.
- *Ignored Phrase (IP)*: A phrase in which the head of the phrase is not connected to the verb in the dependency parse tree. IPs will not be involved in the extraction process.
- *Linked Phrase (LP)*: A phrase that all of its words are linked to one and only one entity.
- *Candidate Phrase (CP)*: A phrase that includes at least one verb or noun. Phrases that do not include any name or verb will not be ignored in triple extraction.

In our unsupervised RTE, triples are extracted based on the following criteria:

- Sentences with no VP are ignored.
- If the sentence has one VP and two LPs, the triple (LP_1, VP, LP_2) is extracted.
- If the sentence has one VP, one LP and more than one CPs (N > 1), we extract N triples as (LP, VP, CP_i) where $i \epsilon (1..N)$ and CP_i is the i^{th} candidate phrase.

The precision of our unsupervised RTE is 67%, evaluated on 600 sentences. To facilitate the verification by human experts, we assign confidence values to each of the extracted triples with a method similar to [33].

6. Mapping

The triples are extracted from multiple heterogeneous sources. As a result, semantically equivalent predicates may be extracted in different lexical forms. For example, "محل تولد", "place of birth" and "birthplace" have the same meaning, but are extracted from different sources. Therefore, it is essential to have a homogenization process on the predicates to map the equivalent lexical forms into a common IRI. Mapping is originally suggested by DBpedia [1] to map Wikipedia infoboxes to the ontology classes and properties.

6.1. Mapping Wikipedia Infoboxes

Wikipedia infoboxes contain predicates in different syntactic shapes and different languages. While DBpedia uses a markup language for mapping, FarsBases uses mapping tables which are easier to maintain by the human experts. In the following, we describe the two approaches.

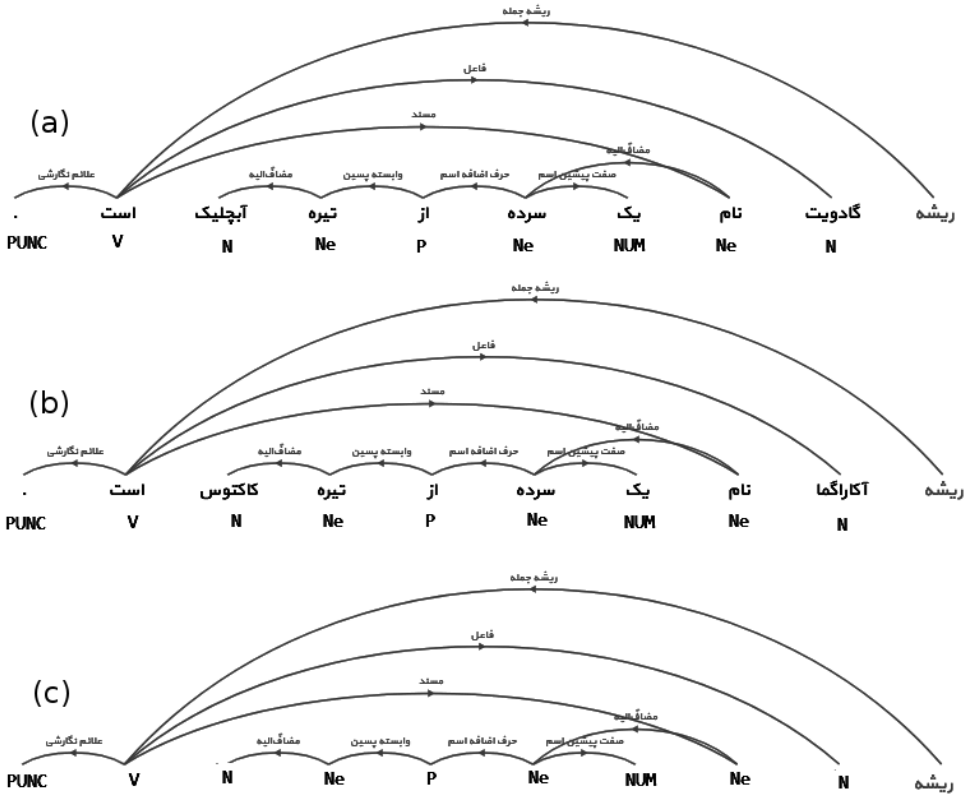

Fig. 2. Two sentences (a,b) with the same dependency pattern (c).

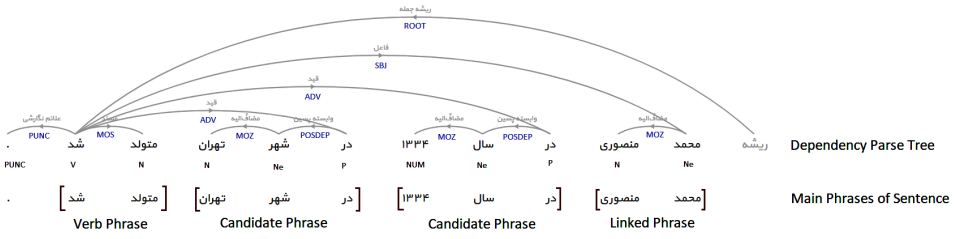

Fig. 3. The dependency parse tree and main phrases of a sample Persian sentence

6.1.1. The DBpedia Approach

Lehmann et al. [23] proposed an approach to map Wikipedia infobox data to the triples based on a markup language that specifies the mapping. DBpedia mappings consist of 1) the mapping of infoboxes to the ontology classes and 2)the mapping of the attributes in the infoboxes to the ontology predicates[6].

6.1.2. Enhancements on DBpedia

Mapping Tables. The main difference between FarsBase and DBpedia approaches is on their mapping representation. While the markup language in DBpedia is

[6]http://mappings.dbpedia.org/

more flexible than the table-based scheme in FarsBase, we observed that it is hard for the human experts to define, read and maintain the markup entries. More importantly, working with markups takes more time and drastically increases the cost of KG construction. FarsBase provides a user interfaces that is more friendly to the experts who are not familiar with markup languages.

Mapping suggestion. In Persian Wikipedia, some of the infobox types and their attributes are written in English but are displayed in Persian in the user interface. Since the English types and attributes are mostly found in the DBpedia mappings, we transfer these mappings to the FarsBase mapping tables. For those types and attributes that are written in Persian, we use *machine translation*. By looking at the Persian attributes, the system automatically translates them into English and looks for their equivalence in English DBpedia. Finally, the human experts need to confirm the suggested mappings. Therefore, this process is semi-automatic.

Transformers. FarsBase uses a novel technique called the *transformer functions* to map complex strings in infobox attributes to multiple pairs of values and RDF types. Transformer functions automate data cleansing and normalization and require minimal effort by the experts. The technique is described in section 6.2.

Mapping for Raw Text Extraction. We propose an *ontology-aware* method for mapping RTE triples in section 6.3. In this method, the mapping entries that are originated from Wikipedia infoboxes are also used to map RTE-generated triples.

6.1.3. Ontology Construction

Ontology is one of the most important parts of any knowledge graph. In FarsBase, each entity is an instance of one of the classes. The FarsBase ontology is a tree of the classes, each of which has only one parent.

For Example سی_و_سه_پل(Si-o-se-pol) is a Bridge, ArchitecturalStructure, Infrastructure, RouteOfTransportation, Place and Thing. The main class of each entity (i.e. the deepest class in the tree) is defined by an fbo:instanceOf predicate.

```
fbr:سی_و_سه_پل fbo:instanceOf
fbo:Bridge .
```

Note that fbr is the prefix for the "http://fkg.iust.ac.ir/resource/" namespace and fbo is the prefix for the "http://fkg.iust.ac.ir/ontology/" namespace. All classes of an entity are defined with rdf:type predicate:

```
fbr:سی_و_سه_پل rdf:type fbo:Bridge ;
    rdf:type fbo:ArchitecturalStructure ;
    rdf:type fbo:Infrastructure ;
    rdf:type fbo:RouteOfTransportation ;
    rdf:type fbo:Place ;
    rdf:type fbo:Thing .
```

The ontology of FarsBase consists of 767 classes. The main body of the ontology is derived from DBpedia. Classes with no entities in Persian Wikipedia, most of which are not extensively used in Persian language and culture, are removed from the ontology. Also, some new classes were found in Persian Wikipedia, which are added to the ontology. The new classes are listed in Table 4.

Experts added some valuable information to this ontology, e.g. different labels for classes, predicates, ranges and domains. FarsBase automatically estimates the range and domain of each property, and stores them in the KG as triples of type fbo:autoRange and fbo:autoDomain.

6.2. FarsBase Mapping and Transformer Functions

In Wikipedia templates and infoboxes, the values are written in different formats. Interpreting each format requires implementing a small logic. Consider the following examples:

- If the infobox type of an article is "Iranian village", the type of entity is fbo:Village. Moreover, it can be concluded from the infobox type that the village is located in fbr:Iran.
- If the value of "years of activity" in an infobox is "1357-1366", this string value should be parsed and converted to two numerical values as the start and end years of the activity.
- If the area_km2 value in an infobox is "1897", the string value should be parsed as a number, and the property is "area". Also, the unit should be set as square kilometre (km^2).
- The ”lat“ (latitude) value for a geographical entities can be either in degrees-minutes-seconds format (e.g. 35 41’ 36") or in numerical signed degrees format like 35.69333333. We have to unify the formats.

To efficiently handle the logic required to handle the values, the experts create a table containing the *mapping rules* for each template. Each rows in the mapping rules table (Table 5) is a *rule* that defines what

Table 4
New classes added to the original DBpedia ontology based on Persian language and culture

New class	Parent class
Mineralogic Zone (Index Mineral)	Mine
Rural District	Governmental Administrative Region
Qanát (Káriz, Underground Canel)	Canal
Waterfall	Stream
Electoral District	District
County	Governmental Administrative Region
Short Story	Book
Marja'	Cleric
Imámzádeh	Religious Building
Scholar (Muslim Scholar)	Scientist
MartialArt	Sport
Seminary (Islamic religious school)	Educational Institution
Imám	Cleric
Militant Group	MilitaryUnit
Intangible Cultural Heritage	Thing
Exchange Market	Organisation

should be done for each of the attributes of a template. For example, a mapping table for the template "a city of Iran" is defined in Table 5.

As an example, suppose that the mapper is given all triples extracted from an article with "a city in the Iran" template. The rules with empty attribute fields are executed at first. Therefore, two triples are generated for type (fbo:City) and location (fbr:Iran). Note that a rule with an empty attribute must have a constant value. In the next step, the mapper checks the infobox attribute in all triples and looks for the corresponding rule in the mapping rules table. For example, if the infobox attribute is "mayor", it will be mapped to "fbo:mayor". Also, wheat_production matches with two rules in the rule set. As a result, several triples corresponding to such attribute are generated.

The transformer functions are defined to handle complex value types. The input of a transformer is a string, and the output can be a string, a numerical value, or a date. For example, the minMaxRangeToMin function receives an interval string (minimum and maximum) and returns the minimum value (Input: "12-13", output: 12). Similarly, the latLong function receives a string corresponding to latitude or longitude, detects its format, and generates a decimal value in signed degrees format (Input: "35 41' 36"", output: 35.69333333).

For some attributes in Wikipedia infoboxes, FarsBase maps the attribute to NULL, meaning that the key will be ignored in the mapping phase and its information will not be stored in the knowledge graph. For example the size of an image in a Wikipedia article is not valuable and is ignored.

Storing mapping data in the rule sets, makes it easier to represent and maintain the mapping system. Manipulating the mapping rules is easy for the experts and provides enough flexibility to handle all complicated cases.

6.3. Mapping Triples from Raw Texts

The Wikipedia mapping system discussed in 6.1 is only applicable to Wikipedia infoboxes. However, if we identify the IRI of the entities, we can also apply the mapping rules on triples extracted from raw text. The module that identifies the entities from the triples generated by RTE and TBE is described in section 5.3.1.

Obviously, identifying the entities and predicates is very challenging due to the ambiguities in triple extraction. To overcome this problem, if the entity has an infobox in Wikipedia, the extracted predicate is matched with an attribute in the infobox according to the mappings of that template. If the attribute is found in the infobox, the triple is mapped according to the Wikipedia mapping. Otherwise, we look for any rules in the mapping-rule tables of all templates that contain the extracted predicate.

To ensure that the information extracted from raw-text have an adequate quality, all RTE-generated triples

Table 5
Mapping rules for Wikipedia templates

Infobox Attribute	Predicate	Constant	Unit/Type	Transformer
-	rdf:type	fbo:City	owl: NamedIndividual	-
-	rdf:location	fbr:Iran	owl: NamedIndividual	-
mayor	fbo:mayor	-	owl:NamedIndividual	
wheat_production	fbo:weathProductionMin	-	xsd:int	minMaxRangeToMin
wheat_production	fbo:weathProductionMax	-	xsd:int	minMaxRangeToMax
area_mi2	fbo:area	-	fbr:km2	mile2ToKm2
latitude	fbo:lat	-	xsd:double	latLong
longitude	fbo:long	-	xsd: double	latLong

must be approved by experts before they can be stored in the Belief-Store.

7. Integration

Once the triples are extracted, they need to be integrated with the KG. This requires different post-processing steps in the integration phase, including the following tasks:

- Mapping Wikipedia infoboxes to the ontology (discussed in section 6)
- Handling N-ary relations
- Versioning and updating the KG
- Human supervision, used for batch verification of the candidate triples
- Linking entities to external datasets

Our integration system is designed such that it can be continuously updated to handle very recent informational queries in the search engine. Moreover, it uses several heuristics for low-cost (batch) verification of the triples in which we have a low confidence.

In this section, we will first explain the mapping process and our initial mapping phase for data triples. Then we show how FarsBase integrates mapped data from multiple sources, how candidate facts are updated and in which process they are promoted to the beliefs.

7.1. N-ary Relations

Most knowledge graphs, FarsBase included, represent data in subject-predicate-object (SPO) format defined by the RDF model. This representation is suitable for relations involving two entities, but it is not straightforward to represent n-ary relations in the SPO format.

Wikipedia infoboxes support n-ary relations by defining an order for attribute-values. Figure 4 shows a sample of n-ary relations in the Wikipedia infobox of a football player. In such cases, all related attributes end by a digit, e.g. years4, caps4 and goals4. Rouces et al. [34] reviewed some modelling patterns for representation of n-ary relations: The basic-triple pattern, triple-reification, singleton-property pattern, specific-role-neo-davidsonia, general-role-neo-davidsonian and Role-class pattern. FarsBase uses triple-reification for handling n-ary relations. For the infobox shown in Figure 4, a sample of the extracted triples is as follows:

```
fbr:علی_دایی fkgo:relatedPredicates
 fbr:علی_دایی/relation_4 .

fbr:علی_دایی/relation_4 rdf:type
 fbo:RelatedPredicates ;
    fbo:mainPredicate fbo:club ;
    fbo:club fbr:پرسپولیس ;
    fbo:caps 38 ;
    fbo:goals 23 .
```

Note that `fbr:علی_دایی/relation_4` is automatically created. The reification model also handles relations that contain temporal data.

7.2. Updating FarsBase

The Mapper system stores all the data generated by the various extraction modules in the CFM-Store. The update system is triggered whenever a new set of triples is generated, either by WKE, RTE or TBE. In each update, the generated files, along with general extraction information (start and end time of the extraction and the extractor module), are stored in the system. The Mapper continuously traverses the arriving updates in short intervals, and runs the mapping process. Note that some of the mapped triples might need

```
{{بخش آغازین کوتاه|اکتبر ۲۰۱۶}}
Infobox football biography}}
name| = علی دایی
Nickname| =
image| = [[پرونده:Ali Daei 2016.jpg|250px]]
caption| =
fullname| =
birth_date = ۱| فروردین ۱۳۴۸ ({{سن|۱۳۴۸|۱|۱|از تاریخ=خورشیدی}})
years4 = ۱۳۷۳–۱۳۷۵|
clubs4| = [[باشگاه فوتبال پرسپولیس|پرسپولیس]]
caps4 = ۳۸|
goals4 = ۲۳|
years5 = ۱۳۷۵–۱۳۷۶|
clubs5| = [[باشگاه ورزشی السد|السد]]
caps5 = ۱۶|
goals5 = ۱۰|
years6 = ۱۳۷۶–۱۳۷۷|
clubs6| = [[باشگاه فوتبال آرمینیا بیله‌فلد آلمان|آرمینیا بیله‌فلد]]
caps6 = ۲۵|
goals6 = ۱۷|
```

علی دایی

شناسنامه	
زادروز	۱ فروردین ۱۳۴۸ (۴۹ سال)
باشگاه‌های حرفه‌ای*	
سال‌ها	باشگاه‌ها	بازی†	(گل)†
۱۳۷۳–۱۳۷۵	پرسپولیس	۳۸	(۲۳)
۱۳۷۵–۱۳۷۶	السد	۱۶	(۱۰)
۱۳۷۶–۱۳۷۷	آرمینیا بیله‌فلد	۲۵	(۱۷)

Fig. 4. N-ary relations in a Wikipedia infoboxe

to be validated before transferring to the final data. The *Update Handler* transfers the mapped triples to the final repository when the data is validated by experts, or when their confidence is above a pre-defined threshold.

Triples that no longer exist in a new Wikipedia dump will be deleted from the final data (Belief-Store). FarsBase stores the version number along with the triples, so that it can handle the changes in the values of the triples. After finding new triples from a module, the mapper uses a version number generated by the CFM-Store and stores all triples with the assigned version to the CFM-Store.The final (validated) data is stored in Belief Store in specific periods (currently each 24 hours).

7.3. Human Supervision

Automatic approaches for knowledge graph construction, such as NELL [2], have recently reached considerably high accuracy, but to construct a high-quality KG, NELL uses crowdsourcing to verify the extracted facts. Therefore, human supervision is still essential to ensure the correctness, consistency, and completeness of the data.

In FarsBase, Human experts are used in four tasks: triple approval, ontology construction, mapping, and evaluation (e.g. to create gold data for evaluating the distant supervision module).

Triples with the confident higher than a threshold are automatically transferred to the Belief-Store. Other triples must be verified by majority vote of three experts.

Since FarsBase is used in a search engine, the cost of a wrong triple is proportional to its popularity in user queries, i.e. its frequency in the user query logs. To minimize this cost, entities are prioritized by their *expected query frequency*, which is estimated by a probabilistic model.

7.4. Linking to External Data Sets

DBpedia is already linked to 36 other datasets. Therefore, we can easily link FarsBase to other datasets if we correctly match the FarsBase entities with DB-

Table 6
Frequency of entities in each classes

Class	Frequency of entities
Settlement	69,317
Village	49,512
Person	26,035
Species	24,350
Historic Place	21,881
Film	19,093
Ship	18,356
Soccer Player	15,180
Planet	11,814
Airport	11,148
Actor	8,106
Music artist	7,227
Chemical Compound	6,290
Office holder	5,900
Canal	5,520
City	5,301

Table 7
Frequency of triples in each classes

Class	Frequency of triples
Settlement	2,965,740
Village	1,155,247
Soccer Player	670,884
Film	610,592
Historic Place	566,268
Species	558,861
Person	476,467
Ship	405,828
Planet	342,807
Airport	282,147
Chemical Compound	273,666
Musical Artist	201,786
Office holder	196,276
Actor	191,179
City	155,559
Canal	152,603

pedia. Most Persian Wikipedia articles have interlanguage links with an English article, and the majority of English articles are mapped to DBPedia. By following the connections, FarsBase is connected to DBpedia using the owl:sameAs predicate. 439,445 of FarsBase entities have at least one link to external datasets. FarsBase totally has 5,582,589 links to external datasets.

8. Evaluation and Statistics

We evaluated FarsBase from different aspects, including the size, precision, mapping quality, coverage, and freshness (timeliness), and the number of links to other datasets.

8.1. Size

Tables 6 and 7 show the frequencies of entities and triples for the most frequent classes in FarsBase. The Settlement and Village classes contain the largest number of entities (69317 and 49512).

In general, more than 190,000 entities are instances of the Place class. The frequency of entities and triples of the first level of the ontology are shown in Table 8.

8.2. Mapping

There are 1712 unique templates in Persian Wikipedia, out of which 683 templates are mapped to FarsBase ontology classes. Also, Persian Wikipedia infoboxes have 25032 unique attributes, and 7808 attributes are mapped to the FarsBase ontology. Table 9, illustrates the number of mapped templates, attributes and Wikipedia infobox triples. While we have only mapped 40% of the templates and 31% of the attributes, the mapping covers 90% of the triples because the unmapped templates and frequencies have a quite lower frequency in the triples.

8.3. Coverage of Entities

We adopted several measures to evaluate the coverage of entities in FarsBase. The coverage is evaluated using two test sets: Wikipedia articles, entities from a gazetteer, and the information need of search engine based on a query log.

Gazetteer Named Entities. We aim to measure how much of the named entities that appear in raw texts are available in FarsBase. We used three gazetteers for named entity recognizer (NER) in Persian developed by Iran Telecommunication Research Center (ITRC). We measured how many of the gazetteer entities are available in FarsBase. Table 10 shows the results for the person, location and organization classes. Results show that FarsBase has a wider coverage for famous persons compared to other named entities.

Query Log Entities. When a KG is used for semantic search, entities that appear in frequent queries are more

Table 8
The frequency of entities and triples of the first level of the ontology

Class	Frequency of Entities	Frequency ofTriples
Place	196,073	6,329,219
Agent	106,669	2,951,735
Work	35,035	1,133,014
Species	24,465	561,777
Mean of Transportation	22,944	539,869
Chemical Substance	7,630	322,617
Event	2,942	118,451
Device	1,403	44,974
Disease	1,306	28,594
Time Period	874	9,883
Language	571	17,666
Food	423	7,838
Flag	364	6,425
Award	344	8,683
Currency	290	7,181
Topical Concept	225	7,528
Holiday	129	3,360

important. We used 321 popular queries from a search engine log. Human experts chose the main query for each entity and checked whether the entity exists in FarsBase. FarsBase covered 91.85% of the frequent entities.

8.4. Triple Coverage for Semantic Search

Application-specific Coverage The search system and its evaluation is beyond the scope of this paper. Nonetheless, it is important how FarsBase covers the triples that are required to answer KG-based semantic queries. Evaluating application-specific triple coverage is very challenging because it depends to the application (KG search engine) on how many triples it needs to have for answering to each query. For example, consider the query پلهای اصفهان("Bridges in Isfahan"). For any of the results of this query, such as the Si-o-se-pol bridge, at least two triples must be available in the KG so that the result can retrieved, namely "result rdf:type fbo:Bridge" and "result fbo:locatedInArea Isfahan". From the application's perspective, if any of the two triples are not in the KG, the cost is the same as not having both of the triples. Therefore, we use an evaluation measure in the application's domain, such as precision, which indirectly measures the triple coverage.

Table 9
Statistics about mapped templates, attributes and triples

	Total count	Mapped count	Percent
Template	1,712	683	40%
Attributes	25,032	7,808	31%
Infobox Triples	11,917,420	10,682,099	90%

Table 10
Coverage of the gazetteer named entities

Named entity class	Count	Coverage
Person	1,771	95.65
Location	4,697	89.71
Organization	5,524	76.64
All	11,992	85.31

Eliminating Application-specific Errors. To have a precise estimate of the triple coverage, we try to eliminate the aspects of the application (i.e. semantic search engine) that are irrelevant to triple coverage. A KG-based semantic search engine can fails to retrieve a result for four reasons:

1. The query is not KG-compatible, e.g. "What to do in case of fire?".
2. The query is not supported by the KG. Note that semantic search engines, like Google knowledge graph, do not try to support all KG-compatible queries because this will reduce the precision. Similarly, the FarsBase query engine only answers a query if it matches one of the pre-defined query templates.

3. The query is not correctly understood by the engine, e.g. it fails to detect entities from the query or to disambiguate the detected entities.
4. Any of the triples required to retrieve the result are not in the KG or are wrong.

To measure the application-specific triple coverage, we have to eliminate the effect of the first three issues. Therefore, the experts carefully revised the query set and chose the KG-compatible queries that are already supported and processed correctly by the query engine.

Query Set. Three experts evaluated the responses by the *semantic search module* for each query

The query set is built from three different sources:

- Expert queries: 62 queries were proposed from three experts.
- Search engine log: 63 queries were selected randomly from the end user query log. If the answer of a query was not in the knowledge graph (checked by human experts), the query was replaced by another query.
- Class-centric queries: 253 queries were selected based on the ontology class of the main entity that appears in the query. For example, in "Bridges in Isfahan", the main entity is "Isfahan" and it belongs to the fbo:City class. These queries were constructed automatically from the triples of each entity.

Results. We used r-precision to measure the application-specific triple coverage. For each query, r-precision is the number of correct results among the first R results, where R is the number of relevant results. For examples, since there are 6 bridges in Isfahan, if the query engine returns only 5 correct results, its r-precision is $\frac{5}{6}$. Average r-precision is the average of the r-precision values for all queries in the query set. Average r-precision for each source is reported in Table 11. Note that, since we have eliminated the errors of the query engine when creating the evaluation query set, r-precision is very close to the recall. The difference is that if an irrelevant result appears in the result set (due to false information in the triples), it will contribute to a lower r-precision but does not affect the recall.

Table 11
R-Precision for different query sets

Source	R-Precision
Expert queries	97.5
Search engine queries	93.2
Class-centric queries	93.5

8.5. Data Freshness

Freshness measures the effectiveness of the update process. To evaluate the freshness of data in FarsBase, a list of famous recent events was collected in a time interval. The list was extracted automatically from “Portal:Current_events” and “Deaths_in_2018” pages in Wikipedia. After checking 100 entities from the list, all the entities and 91.02% of their attributes existed in FarsBase.

8.6. Linked Data

A total of 558,2589 FarsBase resources has been linked to 33 external datasets. In total, 439,445 of the 541,927 FarsBase entities (81.09%) were connected to at least one other datasets. Table 12 shows the number of links to each of the linked datasets.

9. Related Work

9.1. Knowledge Graph Construction

FreeBase. Developed by Metaweb, Freebase was a publicly available knowledge graph launched in early 2007. It contained more than 125,000,000 tuples, 4,000+ types, and 7,000+ properties [39]. Freebase data was harvested from multiple sources including Wikipedia and was available using Metaweb Query Language (MQL).

YAGO. Suchanek et al. [40] presented YAGO as a lightweight, high-quality and extensible ontology. YAGO combined Wikipedia infoboxes and WordNet taxonomic relations using several heuristics and a quality control system. Facts are extracted by applying different heuristics on infoboxes, types, words, and categories. YAGO2 added temporal and spatial aspects to the knowledge graph. To provide geographical information, YAGO2 included the GeoNames database [41]YAGO3 is built using multilingual information extraction techniques and supports 10 languages [3, 42].

Table 12
Datasets linked to FarsBase

Dataset	URI Prefix	Count
All Links	-	5,582,589
All DBpedia	dbpedia.org	3,802,782
English DBpedia	dbpedia.org	652,773
Wikidata	wikidata.org	637,062
YAGO	yago-knowledge.org	410,538
Freebase	freebase.com	397,261
GeoNames	geonames.org	113,314
Deutsche National Bibliothek	d-nb.info	64,378
LinkedGeoData	linkedgeodata.org	31,413
Virtual International Authority File (VIAF)[35]	viaf.org	24,252
EU Open Data	data.europa.eu	22,999
GeoVocab	geovocab.org	15,021
OpenCyc	cyc.com	13,783
rdfabout.com	rdfabout.com	7,723
fu-berlin.de (Drugbank, Sider, WikiCompany, Factbook ...)	fu-berlin.de	6,250
isprambiente.it	isprambiente.it	5,718
GeoSpecies	geospecies.org	5,094
LinkedOpenData	linkedopendata.org	4,881
New York Times	nytimes.com	4,784
LinkedMDB	linkedmdb.org	4,565
Musicbrainz[36]	zitgist.com	4,080
European Nature Information System	eunis.eea.europa.eu	1,675
BBCWildlife Finder	bbc.co.uk	1,593
Linked Open Data of Ecology (LODE)[37]	ecowlim.tfri.gov.tw	959
Camera dei deputati	camera.it	675
The LinkedWeb APIs ontology [38]	linked-web-apis.fit.cvut.cz	347
Dublin Core Metadata Initiative	purl.org	345
OpenEI	openei.org	304
270a Linked Dataspaces	270a.info	299
Eurostat (Linked Stats)	eurostat.linked-statistics.org	187
GHO	ghodata	140
logainm.ie	data.logainm.ie	88
UK Learning Providers	id.learning-provider.data.ac.uk	58
DBTune	dbtune.org	17
Revyu	revyu.com	4

DBpedia. DBpedia converted Wikipedia contents into a large multi-domain RDF dataset. It is interlinked to other open data sources including FOAF, GeoNames, Dublin Core Berlin, World Factbook, and Music Brainz. The DBpedia community also developed a series of modules which makes DBpedia accessible via Web services [43]. DBpedia has been further developed into a multilingual knowledge graph and aims to completely support all languages in Wikipedia. The DBpedia extraction framework contains four extraction modules: Mapping-Based Infobox Extraction, Raw Infobox Extraction, Feature (e.g. geographic coordinates) Extraction, Statistical Extraction. Using the mapping-based infobox extraction system [1, 44].

BabelNet. BabelNet integrated lexicographic and encyclopedic knowledge across multiple languages and presented a lightweight method to map Wikipedia articles to WordNet senses. For resource-poor languages, they used human-edited and statistical machine translations of Wikipedia articles in the other languages.

BabelNet also integrated a multilingual word-sense disambiguation system with its knowledge graph [6].

Wikidata. The goal of Wikidata (Wikipedia for data) is managing the factual information of Wikipedia. Wikidata uses facts instead of triples and users enter facts directly to the database. Each entity has an ID (not a URI). There is no extraction process on Wikidata and each fact are entered by the users. Each entity has multiple "statements" instead of "triples". Each statement has one or more references and one claim[45, 46].

Other Knowledge Graphs. Wang et al. [47] proposed an automatic knowledge graph construction using statistical text over database systems (MADDEN), deductive reasoning system (PROBKB) and human feedback (CAMEL).

Knowledge Vault introduces a probabilistic knowledge graph that mixes information extraction from Web content with prior knowledge derived from existing KGs. This knowledge graph uses a supervised machine learning method to fuse these information sources. Knowledge Vault has tree main components: triple extractors (text documents, HTML trees, HTML tables and Human annotated pages). Graph-based priors, and knowledge fusion. Triple extractors may extract unreliable and noisy knowledge. Prior knowledge is used to reduce the noise from the extracted data [48].

DeepDive cleans and integrates data from multiple sources like text documents, PDFs and structured databases. Statistical inference and machine learning is used to extract tuples and defines a probability score for each of tuple[49, 50].

Nguyen et al. [51] argued that most of the knowledge graphs are not extracted based on the information needs of the end users and fail to cover many relevant predicates. They suggested QKBfly, an approach to construct an on-the-fly knowledge graph from user's queries a question answering system.

FrameBase [34] proposed a KG schema which uses FrameNet [52] to store and query n-ary relations (facts with more than two entities or literals) from heterogeneous sources which combine efficiency and expressiveness. The article investigated different triple representations for n-ary relations and used NLP frames to handle this relations on other knowledge graphs.

Comparative Survey. Farber et al. defined 35 aspects in seven categories, including general information, format and representation, genesis and usage, entities, relations, schema and particularities. for a comparative study on knowledge graphs they provide a thorough comparison of these aspects in DBpedia, Freebase, OpenCyc, Wikidata, and YAGO [53].

KG Construction for other Low-resource Languages. Persian is originated from the ancient Middle-Persian language and has an extensive vocabulary derived from the Classic Arabic language. Persian and Arabic come from different roots (i.e., Indo-European and Semitic), but Persian script is an adaption of Arabic script with a few modifications. Currently, DBpedia has an Arabic chapter[7] [54–56]. While FarsBase has mapped 683 of 1712 Wikipedia templates for Persian (see Table 9), DBpedia mapping wiki for Arabic[8] contains only 67 infobox types. In 2017, Ktob and Li[57] introduced a preliminary version of the Arabic Knowledge Graph (AKG). The paper investigated the challenges and opportunities for constructing AKG. The data is publicly available. They enumerated some challenges concerning the volume of the data and lack of essential tools for constructing AKG. They also introduced Marifa, an open source tool that extracts information from Arabic catalogues in the Excel format and converts it to RDF triples. Marifa contains three modules, namely the Excel Parser, The Extractor, and The Discovery module. The discovery module attempts to link the extracted triples to three external data sources including the Arabic chapter of DBpedia, the English DBpedia, and Wikidata. To our knowledge, there is no existing knowledge graphs specifically created for other low-resource languages such as Indian, Urdu and Turkish.

9.2. Quality Assessment

Paulheim et al. [58] provided a survey on knowledge graph refinement approaches (completion vs. error detection, target of refinement, and internal vs. external methods) and evaluation methodologies (partial gold standards, knowledge graph as silver standards, retrospective evaluation, and computational performance). They evaluated Cyc [59] and OpenCyc, Freebase, DBpedia, YAGO, NELL and Knowledge Vault. They also presented some statistics about Wikidata, Google's Knowledge Graph, Yahoo!'s Knowledge Graph[60], Microsoft's Satori and Facebook Entity Graph.

Färber et al. proposed 34 metrics for knowledge graph quality assessments and analyzed on the DBpedia, Freebase, OpenCyc, Wikidata, and YAGO [4]. The

[7] https://wiki.dbpedia.org/join/chapters
[8] http://mappings.dbpedia.org/index.php/Mapping_ar

metrics categorized on Intrinsic (accuracy, trustworthiness and consistency), Contextual (relevancy, completeness, timeliness), Representational Data Quality (ease of understanding and interoperability) and Accessibility (accessibility, license, interlinking) metrics.

Rashid et al. [61] also proposed 4 quality assessments (Persistency, Historical Persistency, Consistency, and Completeness) and assessed this metrics on 11 release of DBpedia and 8 release of 3cixty [62].

9.3. Mapping

Dimou, et al. [63] proposed a uniform assessment approach using the RML (a mapping language) and RDFUnit (a test-driven approach for every vocabulary, ontology or dataset). They also applied this validation on DBpedia [64].

Ahmeti, et al. [65] proposed using the DBpedia mapping infrastructure to enhance Wikipedia content using an Ontology-Based Data Management (OBDM) approach, for example, using conflict resolution policies to ensure the consistency of updates on Wikipedia infoboxes.

9.4. Relation Extraction from Raw Texts

Entity Extraction and Entity Linking. Many researchers have worked on triple and relation extraction from raw texts. In knowledge graph construction, we have to use triple extraction and each entities must be linked to our knowledge graph. Thus entity linking is an essential task in triple extraction.

Han et al. [66] introduced collective entity linking which works based on jointing name mentions in the same document by a representation called Referent Graph.

Exner et al. [67, 68] proposed an entity extraction pipeline which includes a semantic parser and coreference resolver and worked based on coreference chains. This approach extracted more than 1 million triples from 114000 Wikipedia articles. Oramas et al. [69] presented a rule-based approach for extracting knowledge from the songfacts.com website. The extraction pipeline includes Babelfy [70] as a state-of-the-art entity linker with highest precision on musical entities. Nguyen et al. [71] presented J-REED as a joint approach for entity linking based on graphical models. Finally, Röder et al. [72] proposed a GERBIL including an evaluation algorithm for entity linking to compare two entity URIs without being bound to a specific knowledge graph.

Triple and Relation Extraction. Bach et al. [73] presented a review on most important supervised and semi-supervised relation extraction prior to 2007. Kasneci et al. [13] proposed YAGO NAGA which extracts candidate facts from raw text and integrates them to YAGO. NAGA also employs a consistency checking approach to control the quality of generated facts. Wanderlust [74] is an approach to extract semantic relations from raw texts unsupervised dependency grammar approach.

Nakashole [75] presented automatic extraction of a web-scale knowledge graph. He proposed a robust method to extract high quality facts from noisy text sources. He also proposed a method to handle new entities in the dynamic web sources.

TokenRegex [31] which was proposed by Stanford Natural Language Processing Group, facilitate rule-based approaches for relation extraction by implementing a framework for cascadeding regular expressions over sequences of tokens. Madaan et al. [76] proposed a relation extractor for numerical data (e.g. atomic number like “Aluminium, 13” or inflation rate like “India, 10.9%”) which tries to handle units with minimal human supervision.

Presutti et al. [77] proposed Legalo, an unsupervised open-domain knowledge extractor. Legalo is based on the hypothesis that hyperlinks between two entities provide semantic relations between them.

Speer et al. [78] proposed ConceptNet, a knowledge graph extracted from many sources including Open Mind Common Sense (OMCS) [79], Wiktionary, purposeful games, WordNet, JMDict [80] (a Japanese-multilingual dictionary), OpenCyc, and a subset of DBpedia.

Vo et al. [81] described second generation of OIE which is able to extract relations between Noun and Pre, Verb and Prep using deep linguistic analysis. Stanovsky et al. [82] presented a supervised sequence tagging approach to OIE which is aimed by Semantic Role Labeling models. To provide training data, they automatically converted a question answering dataset to an open IE corpus.

Distant Supervision. Distant supervision (DS) was first introduced by Mintz et al. [32]. They examined DS on Freebase and the model extracted 10,000 instances and 102 relations with a precision of 67.6%. Aprosio et al. [10] used DS for extracting missing values from Wikipedia articles to enrich DBpedia knowledge graph. Augenstein et al. [83] uses DS on Web sources with data sparsity, noise and lexical ambiguity.

To handle such data, they used an entity recognition tool and an unsupervised co-reference resolver. They also presented several methods for information integration to aggregate extracted knowledge to the main KG. Heist et al. [84] proposed a language-independent approach based on distant supervision and extracted 1.6 million triples with 95% precision from the abstract of 21 Wikipedia language editions.

9.5. Knowledge Graph Augmentation

FACT EXTRACTOR [85] is an n-ary relation extractor which uses FrameNet and populates the KG with supervised NLP layers and tried to reduce the supervision costs by crowdsourcing. Chen et al. [86] proposed a parallel first-order rule mining method (Path-finding algorithm) and a pruning system to augment Freebase and mined 36625 rules and 0.9 billion new facts from Freebase.

9.5.1. Using Web Tables for Knowledge Augmentation

Limaye et al. [25] proposed machine learning techniques to find entities and type of entities in Web tables and extract relation type from the tables.

InfoGather[19] is an entity augmentation which works on entity-attribute tables. This information gathering system, works based on three core operations: Augmentation by attribute name, Augmentation by example, and Attribute discovery. InfoGather crawls, extracts and identifies relational tables from the Web and builds a table graph from them.

TabEL [16] is a entity linking tool especially designed for Web tables. Assuming that a set of entities or relations in a table are in a particular type, TabEL assigns higher likelihood to entities with higher co-occurrence in Wikipedia articles.

Ritze et al. worked on Web table matching to find missing triples on the knowledge graphs. They proposed T2K Match framework [17] to match triples in Web Tables Corpus (147 million tables) with a knowledge graph, and used this matching tables to augment DBpedia as a cross-domain knowledge graph [18].

10. Conclusion and Future Work

In this paper, We presented FarsBase, a multi-source knowledge graph that is specifically designed to provide linked data for answering semantic queries in Persian search engines. It leverages multiple engines for extracting knowledge from different types of data sources such as Wikipedia, Web-tables and unstructured texts. The extracted triples are then mapped to the ontology and integrated with the main Belief-Store. We adopted two policies for designing FarsBase. First, the extracted information should be up-to-date to cover queries about recent facts, hence FarsBase several different heuristics to automate the updates. Second, the triples must have enough accuracy to prevent returning wrong results, specifically for the frequent queries in semantic search engine. This is addressed by leveraging the query log and various cost models. Our approach also helps maintaining an affordable cost for human supervision, e.g. using rule-based methods for information extraction, and batch-verification of triples by human experts.

We identified several ideas for improving FarsBase. Future work will include:

Linking to FarsNet. FarsNet [87] is a valuable lexical dataset in Persian. It provides a clean and carefully handcrafted hierarchy of concepts. Despite that we have already linked parts of FarsBase with FarsNet, a complete linking between the two sources can increase the extendability of both datasets, and benefits certain applications for the Persian language. Interestingly, many of the FarsNet entities and predicates are already mapped to WordNet synsets, which could pave the path for designing multi-lingual applications.

Integrating with Other Persian Sources. Using structural datasets such as Persian wikis, theses, academic information, and books, can help to further augment the FarsBase data.

OWL Reasoning. Reasoning is an effective methods for augmentation on FarsBase [88–91]. OWL supports standard reasoning capabilities, including symmetric, inverse, transitive, and functional properties. Triples produced by reasoning can be verified by the experts.

Deep Learning for Raw Texts. The idea of applying deep learning for information extraction is becoming popular. Deep learning can be applied as an independent extraction tool or can be integrated with the current stack using frameworks like Snorkel [14] and DeepDive [50].

Temporal and Spatial Aspects. FarsBase currently extracts temporal information from the infoboxes and exploits them to model n-ary relations. However, temporal information can also be extracted and enhanced using other sources such as Wikipedia categories. Sim-

ilarly, Integrating the spatial information and geographical location of the entities to FarsBase can benefit location-based applications.

Codes and Data

All source codes are available at https://github.com/IUST-DMLab/farsbase-kg under Apache License 2.0. The Github page also includes links to the data files, the SPARQL Endpoint, the semantic search engine, and the sample resource IRIs.

Acknowledgements

This research is funded by the Iran Telecommunication Research Center (ITRC). We thank all FarsBase contributors, including Mohammad-Bagher Sajadi, Hossein Khademi-Khaledi, Mostafa Mahdavi, Abolfazl Mahdizadeh, Nasim Damirchi, Leila Oskooei, Ensieh Hemmatan, Morteza Khaleghi, Razieh FarjamFard, Mohammad Abdous, Mohsen Rahimi, Ehsan Shahshahani, Yousef Alizadeh, and Dr. Samad Paydar. We also thank the editors and the reviewers for their valuable comments.

References

[1] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer et al., DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia, *Semantic Web* **6**(2) (2015), 167–195.

[2] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves and J. Welling, Never-Ending Learning, in: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.

[3] F. Mahdisoltani, J. Biega and F. Suchanek, Yago3: A knowledge base from multilingual wikipedias, in: *7th Biennial Conference on Innovative Data Systems Research*, CIDR Conference, 2014.

[4] M. Färber, F. Bartscherer, C. Menne and A. Rettinger, Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago, *Semantic Web* (2016), 1–53.

[5] L. Ehrlinger and W. Wöß, Towards a Definition of Knowledge Graphs., in: *SEMANTiCS (Posters, Demos, SuCCESS)*, 2016.

[6] R. Navigli and S.P. Ponzetto, BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, *Artificial Intelligence* **193** (2012), 217–250.

[7] B. Vatant and M. Wick, Geonames ontology, *Dostupné online:< http://www. geonames. org/ontology/ontology_v3* **1** (2012).

[8] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel et al., Never-ending learning, *Communications of the ACM* **61**(5) (2018), 103–115.

[9] M. Shamsfard, Challenges and open problems in Persian text processing, *Proceedings of LTC* **11** (2011).

[10] A.P. Aprosio, C. Giuliano and A. Lavelli, Extending the Coverage of DBpedia Properties using Distant Supervision over Wikipedia., in: *NLP-DBPEDIA@ ISWC*, 2013.

[11] E. Munoz, A. Hogan and A. Mileo, Triplifying Wikipedia's Tables., *LD4IE@ ISWC* **1057** (2013).

[12] E. Muñoz, A. Hogan and A. Mileo, Using linked data to mine RDF from wikipedia's tables, in: *Proceedings of the 7th ACM international conference on Web search and data mining*, ACM, 2014, pp. 533–542.

[13] G. Kasneci, M. Ramanath, F. Suchanek and G. Weikum, The YAGO-NAGA approach to knowledge discovery, *SIGMOD Rec.* **37**(4) (2008), 41–47.

[14] A. Ratner, S.H. Bach, H. Ehrenberg, J. Fries, S. Wu and C. Ré, Snorkel: Rapid training data creation with weak supervision, *Proceedings of the VLDB Endowment* **11**(3) (2017), 269–282.

[15] P. Li, H. Wang, H. Li and X. Wu, Employing Semantic Context for Sparse Information Extraction Assessment, *ACM Transactions on Knowledge Discovery from Data (TKDD)* **12**(5) (2018), 54.

[16] C.S. Bhagavatula, T. Noraset and D. Downey, TabEL: entity linking in web tables, in: *International Semantic Web Conference*, Springer, 2015, pp. 425–441.

[17] D. Ritze, O. Lehmberg and C. Bizer, Matching html tables to dbpedia, in: *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, ACM, 2015, p. 10.

[18] D. Ritze, O. Lehmberg, Y. Oulabi and C. Bizer, Profiling the Potential of Web Tables for Augmenting Cross-domain Knowledge Bases, in: *Proceedings of the 25th International Conference on World Wide Web - WWW '16*, International World Wide Web Conferences Steering Committee., 2016, pp. 251–261.

[19] M. Yakout, K. Ganjam, K. Chakrabarti and S. Chaudhuri, Infogather: entity augmentation and attribute discovery by holistic matching with web tables, in: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, ACM, 2012, pp. 97–108.

[20] A. Moschitti, K. Tymoshenko, P. Alexopoulos, A. Walker, M. Nicosia, G. Vetere, A. Faraotti, M. Monti, J.Z. Pan, H. Wu et al., Question Answering and Knowledge Graphs, in: *Exploiting Linked Data and Knowledge Graphs in Large Organisations*, Springer, 2017, pp. 181–212.

[21] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber and P. Cimiano, Template-based question answering over RDF data, in: *Proceedings of the 21st international conference on World Wide Web*, WWW, 2012, pp. 639–648.

[22] A. Abujabal, M. Yahya, M. Riedewald and G. Weikum, Automated template generation for question answering over knowledge graphs, in: *Proceedings of the 26th international conference on world wide web*, WWW, 2017, pp. 1191–1200.

[23] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer and C. Bizer, DBpedia - A Large-scale, Multilingual Knowl-

edge Base Extracted from Wikipedia, *Semantic Web Journal* **1** (2012), 1–5.
[24] F.M. Suchanek, G. Kasneci and G. Weikum, Yago: A large ontology from wikipedia and wordnet, *Web Semantics: Science, Services and Agents on the World Wide Web* **6**(3) (2008), 203–217.
[25] G. Limaye, S. Sarawagi and S. Chakrabarti, Annotating and searching web tables using entities, types and relationships, *Proceedings of the VLDB Endowment* **3**(1–2) (2010), 1338–1347.
[26] S. Mohtaj, B. Roshanfekr, A. Zafarian and H. Asghari, Parsivar: A Language Processing Toolkit for Persian., in: *LREC*, 2018.
[27] A. Mirzaei and P. Safari, Persian Discourse Treebank and coreference corpus., in: *LREC*, 2018.
[28] K. Dashtipour, M. Gogate, A. Adeel, A. Algarafi, N. Howard and A. Hussain, Persian Named Entity Recognition, in: *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2017 IEEE 16th International Conference on*, IEEE, 2017, pp. 79–83.
[29] F. Liu, H. Lu and G. Neubig, Handling Homographs in Neural Machine Translation, *arXiv preprint arXiv:1708.06510* (2017).
[30] R. Mitkov, Robust pronoun resolution with limited knowledge, in: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, Association for Computational Linguistics, 1998, pp. 869–875.
[31] A.X. Chang and C.D. Manning, TokensRegex: Defining cascaded regular expressions over tokens, *Tech. Rep. CSTR 2014-02* (2014).
[32] M. Mintz, S. Bills, R. Snow and D. Jurafsky, Distant supervision for relation extraction without labeled data, in: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, Association for Computational Linguistics, 2009, pp. 1003–1011.
[33] O. Etzioni, A. Fader, J. Christensen, S. Soderland and M. Mausam, Open information extraction: The second generation., in: *IJCAI*, Vol. 11, 2011, pp. 3–10.
[34] J. Rouces, G. De Melo and K. Hose, FrameBase: Enabling integration of heterogeneous knowledge, *Semantic Web Journal* **8**(6) (2017), 817–850.
[35] B.B. Tillett, A Virtual International Authority File. (2001).
[36] A. Swartz, Musicbrainz: A semantic web service, *IEEE Intelligent Systems* **17**(1) (2002), 76–77.
[37] G.-S. Mai, Y.-H. Wang, Y.-J. Hsia, S.-S. Lu, C.-C. Lin et al., Linked Open Data of Ecology (LODE): a new approach for ecological data sharing, *Taiwan Journal of Forest Science* **26**(4) (2011), 417–424.
[38] M. Dojchinovski and T. Vitvar, Linked web APIs dataset, *Semantic Web* (2018), 1–11.
[39] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, AcM, 2008, pp. 1247–1250.
[40] F.M. Suchanek, G. Kasneci and G. Weikum, Yago: a core of semantic knowledge, in: *Proceedings of the 16th international conference on World Wide Web*, ACM, 2007, pp. 697–706.
[41] J. Hoffart, F.M. Suchanek, K. Berberich and G. Weikum, YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia, *IJCAI International Joint Conference on Artificial Intelligence* **194** (2013), 3161–3165.
[42] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey and G. Weikum, YAGO: a multilingual knowledge bse form Wikipedia, Wordnet and Geonames, in: *In International Semantic Web Conference*, Vol. 94, Springer, 2016, pp. 1–26.
[43] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives, Dbpedia: A nucleus for a web of open data, in: *The Semantic Web*, Springer, 2007, pp. 722–735.
[44] P.N. Mendes, M. Jakob and C. Bizer, DBpedia: A Multilingual Cross-Domain Knowledge Base, *Language Resources and Evaluation LRES* (2012), 1813–1817.
[45] D. Vrandečić and M. Krötzsch, Wikidata: a free collaborative knowledgebase, *Communications of the ACM* **57**(10) (2014), 78–85.
[46] A. Ismayilov, D. Kontokostas, S. Auer, J. Lehmann, S. Hellmann et al., Wikidata through the Eyes of DBpedia, *Semantic Web* (2018), 1–11.
[47] Wang, Daisy Zhe, Yang Chen, Sean Goldberg, Christan Grant and K. Li, Automatic knowledge base construction using probabilistic extraction, deductive reasoning, and human feedback, in: *In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, Association for Computational Linguistics, 2012, pp. 106–110.
[48] 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, *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14* (2014), 601–610.
[49] A. Halevy, Technical Perspective : Incremental Knowledge Base Construction Using DeepDive, in: *SIGMOD Record*, Vol. 45, 2016, p. 2016.
[50] C. Zhang, C. Ré, M. Cafarella, C. De Sa, A. Ratner, J. Shin, F. Wang and S. Wu, DeepDive: Declarative knowledge base construction, *Communications of the ACM* **60**(5) (2017), 93–102.
[51] Nguyen, Dat Ba, Abdalghani Abujabal, Nam Khanh Tran, Martin Theobald and G. Weikum, Query-Driven On-The-Fly Knowledge Base Construction, *Proceedings of the VLDB Endowment* **11**(1) (2017), 66–77.
[52] C.F. Baker, C.J. Fillmore and J.B. Lowe, The berkeley framenet project, in: *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, Association for Computational Linguistics, 1998, pp. 86–90.
[53] M. Färber, B. Ell, C. Menne and A. Rettinger, A Comparative Survey of DBpedia , Freebase,OpenCyc,Wikidata,And YAGO, *Semantic Web* **1** (2015), 1–5.
[54] H. Al-Feel, A Step towards the Arabic DBpedia, *International Journal of Computer Applications* **80**(3) (2013).
[55] A.O. Bahanshal and H.S. Al-Khalifa, Toward Recipes for Arabic DBpedia, in: *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, ACM, 2013, p. 331.
[56] A.S. Ismail, H. Al-Feel and H.M. Mokhtar, Introducing a new arabic endpoint for DBpedia internationalization project, in: *Proceedings of the 20th International Database Engineering & Applications Symposium*, ACM, 2016, pp. 284–289.

[57] A. Ktob and Z. Li, The Arabic Knowledge Graph: Opportunities and Challenges, in: *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, IEEE, 2017, pp. 48–52.

[58] H. Paulheim, Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods, *Semantic Web* (2015), 1–0.

[59] D.B. Lenat, CYC: A large-scale investment in knowledge infrastructure, *Communications of the ACM* **38**(11) (1995), 33–38.

[60] R. Blanco, B.B. Cambazoglu, P. Mika and N. Torzec, Entity recommendations in web search, in: *International Semantic Web Conference*, Springer, 2013, pp. 33–48.

[61] M. Rashid, M. Torchiano, G. Rizzo and N. Mihindukulasooriya, A Quality Assessment Approach for Evolving Knowledge Bases, *Semantic Web Preprint* (2018).

[62] G. Rizzo, R. Troncy, O. Corcho, A. Jameson, J. Plu, J.C.B. Hermida, A. Assaf, C. Barbu, A. Spirescu, K.-D. Kuhn et al., 3cixty@ Expo Milano 2015: Enabling Visitors to Explore a Smart City (2015).

[63] A. Dimou, D. Kontokostas, M. Freudenberg, R. Verborgh, J. Lehmann, E. Mannens, S. Hellmann and R. Van de Walle, Assessing and refining mappingsto rdf to improve dataset quality, in: *International Semantic Web Conference*, Springer, 2015, pp. 133–149.

[64] A. Dimou, D. Kontokostas, M. Freudenberg, R. Verborgh, J. Lehmann, E. Mannens and S. Hellmann, DBpedia mappings quality assessment, *CEUR Workshop Proceedings* **1690** (2016).

[65] A. Ahmeti, J.D. Fernández, A. Polleres and V. Savenkov, Updating wikipedia via DBpedia mappings and SPARQL, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **10249 LNCS** (2017), 485–501.

[66] X. Han, Collective Entity Linking in Web Text : A Graph-Based Method, *Sigir* (2011), 765–774.

[67] P. Exner and P. Nugues, Entity extraction: From unstructured text to dbpedia rdf triples, *CEUR Workshop Proceedings* **906**(Iswc) (2012), 58–69.

[68] P. Exner and P. Nugues, Entity Extraction : From Unstructured Text to DBpedia RDF Triples, in: *The Web of Linked Entities Workshop (WoLE 2012)*, 2012, pp. 58–69.

[69] S. Oramas, L. Espinosa-Anke, M. Sordo, H. Saggion and X. Serra, Information extraction for knowledge base construction in the music domain, *Data and Knowledge Engineering* **106** (2016), 70–83.

[70] A. Moro, A. Raganato and R. Navigli, Entity linking meets word sense disambiguation: a unified approach, *Transactions of the Association for Computational Linguistics* **2** (2014), 231–244.

[71] D.B. Nguyen, M. Theobald and G. Weikum, J-REED: Joint Relation Extraction and Entity Disambiguation, in: *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*, 2017, pp. 2227–2230.

[72] M. Röder, R. Usbeck and A.-C. Ngonga Ngomo, GERBIL– Benchmarking Named Entity Recognition and Linking Consistently, *Semantic Web* (2017), 1–21.

[73] N. Bach and S. Badaskar, A survey on relation extraction, *Language Technologies Institute Carnegie Mellon University www ark cs cmu eduLS2images997BachBadaskar* (2007).

[74] A. Akbik and J. Broß, Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns, *CEUR Workshop Proceedings* **491** (2009), 6–15.

[75] N.T. Nakashole, Automatic Extraction of Facts , Relations , and Entities for Web-Scale Knowledge Base Population, PhD thesis, University of Saarland, 2012.

[76] A. Madaan, A. Mittal, Mausam, G. Ramakrishnan and S. Sarawagi, Numerical Relation Extraction with Minimal Supervision, *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)* (2016), 2764–2771.

[77] V. Presutti, A.G. Nuzzolese, S. Consoli, D.R. Recupero and A. Gangemi, From hyperlinks to Semantic Web properties using Open Knowledge Extraction, *Semantic Web Journal* **7**(4) (2016), 1–5.

[78] R. Speer, J. Chin and C. Havasi, ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, in: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) ConceptNet*, 2017, pp. 4444–4451.

[79] P. Singh et al., The public acquisition of commonsense knowledge, in: *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, 2002.

[80] J. Breen, JMDict: a Japanese-multilingual dictionary, in: *Proceedings of the Workshop on Multilingual Linguistic Ressources*, Association for Computational Linguistics, 2004, pp. 71–79.

[81] D.-T. Vo and E. Bagheri, Open information extraction, *Encyclopedia with Semantic Computing and Robotic Intelligence* **01**(01) (2017), 1630003.

[82] G. Stanovsky, J. Michael, L. Zettlemoyer and I. Dagan, Supervised Open Information Extraction, in: *Proceedings of NAACL-HLT*, Association for Computational Linguistics, New Orleans, Louisiana, pp. 885–895.

[83] I. Augenstein, D. Maynard and F. Ciravegna, Distantly supervised Web relation extraction for knowledge base population, *Semantic Web Journal* **7**(4) (2016), 335–349.

[84] N. Heist and H. Paulheim, Language-agnostic relation extraction from wikipedia abstracts, in: *In International Semantic Web Conference*, Vol. 10587 LNCS, Springer, 2017, pp. 383–399.

[85] M. Fossati, E. Dorigatti and C. Giuliano, N-ary Relation Extraction for Joint T-Box and A-Box Knowledge Base Augmentation, *Semantic Web Journal* **0**(0) (2015), 1–28.

[86] Y. Chen, D.Z. Wang and S. Goldberg, ScaLeKB: scalable learning and inference over large knowledge bases, *The VLDB Journal* **25**(6) (2016), 893–918.

[87] M. Shamsfard, A. Hesabi, H. Fadaei, N. Mansoory, A. Famian, S. Bagherbeigi, E. Fekri, M. Monshizadeh and S.M. Assi, Semi automatic development of farsnet; the persian wordnet, in: *Proceedings of 5th Global WordNet Conference, Mumbai, India*, Vol. 29, 2010.

[88] S. Colucci, F.M. Donini and E. Di Sciascio, Reasoning over RDF Knowledge Bases: where we are, in: *Conference of the Italian Association for Artificial Intelligence*, Springer, 2017, pp. 243–255.

[89] J.-H. Qian, X. Jin, Z.-J. Zhang and C. Shao, Construction of Knowledge Base Based on Ontology, in: *Proceedings of the 2017 International Conference on Wireless Communications, Networking and Applications*, ACM, 2017, pp. 77–83.

[90] Z. Quan and V. Haarslev, A parallel computing architecture for high-performance OWL reasoning, *Parallel Computing* (2018).

[91] S. Batsakis, E.G. Petrakis, I. Tachmazidis and G. Antoniou, Temporal representation and reasoning in OWL 2, *Semantic Web* **8**(6) (2017), 981–1000.
[92] N. Xie, C. Cao and H. Guo, A knowledge fusion model for web information, in: *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, IEEE, 2005, pp. 67–72.
[93] X.L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun and W. Zhang, From data fusion to knowledge fusion, *Proceedings of the VLDB Endowment* **7**(10) (2014), 881–892.
[94] X.L. Dong and D. Srivastava, Knowledge curation and knowledge fusion: challenges, models and applications, in: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM, 2015, pp. 2063–2066.