Deriving of Thematic Facts from Unstructured Texts and Background Knowledge

Nataliya Yelagina and Michail Panteleyev

Saint-Petersburg State Electrotechnical University "LETI", Russia
{natyelagin,mpanteleyev}@gmail.com

Abstract. When developing information-analytical systems (IAS) for various purposes it is often necessary to gather *thematic facts* which are of interest to experts in the field. The paper presents an approach that allows one to increase the completeness of fact extraction by using basic domain knowledge. The main idea of the approach is deriving new facts on the basis of facts explicitly stated in the text and basic knowledge contained in the corresponding ontologies. An architecture and algorithms of the system are discussed. The approach is illustrated by an example of extracting relevant facts using inference rules.

1 Introduction

The Internet and corporate databases store huge amount of unstructured documents. Therefore the problem of automated extraction of relevant information from these documents is attracting the attention of many researchers in the field of Text Mining and Information Extraction. In these areas a lot of approaches and techniques have been proposed many of which are specifically tailored to particular problems that they are meant to address [1].

When developing IAS for decision making support the experts are interested in getting facts characterizing various aspects of the analyzed objects. However, relevant facts are not always mentioned in the analyzed texts explicitly. These facts in some cases can be inferred from the facts contained in the texts and some basic knowledge. This paper presents the approach to solving this task. The paper is organized as follows. Section 2 provides an overview of related work. Section 3 explains the proposed approach and basic models. Section 4 presents the algorithms of the FactE system implementing the approach. In Section 5 implementation details of FactE system prototype are presented. In Section 6 some preliminary experimental results is discussed. Finally, the conclusion discusses the possible directions for improving the system.

2 Related Work

Fact extraction from unstructured text is currently the subject of many works. Some of them have the objective to solve the general task of information extraction while others aim at extracting facts of a more complex structure. Several approaches to fact extraction are known; the ontology-based one is considered in

P. Klinov and D. Mouromtsev (Eds.): KESW 2014, CCIS 468, pp. 208–218, 2014.

this paper. The fundamental work [2] reveals the state-of-the-art of this subfield of information extraction and presents the corresponding systems.

Paper [3] considers the task of extracting facts as RDF-triples by identifying specific instances of the ontology in sentences of the text, and composing RDF-triples, replenishing the ontology. The approach to the problem is based on extracting each of the triple elements by searching the corresponding parse tree. The paper presents the Onto-Text system implementing the approach.

The SOBA system [4] is able to automatically create a knowledge base while analyzing texts. The system allows to process documents from heterogeneous sources – text, tables, and image captions. SOBA includes a webcrawler which allows to find new sources on the subject, linguistic annotation module and mapping module that allows to project the information found in the sources on the ontology elements. Text processing is guided by extraction rules.

Most existing ontology-based information extraction (OBIE) systems allow to identify knowledge, explicitly mentioned in text, by using ontological knowledge while analyzing the documents. In addition to the broad descriptive features ontologies also have the advanced features of inferring knowledge. Some systems actively use reasoning as a partial replacement to the traditional techniques of extracting information. BOEMIE [5], for instance, is a generic content analysis system processing texts, video, audio inputs, etc. Inference in BOEMIE is based on automatically acquired rules that operate information extracted from the source. This system only uses knowledge, explicitly mentioned in the text.

Use and expansion of the accumulated knowledge ontology is implemented in SOFIE system [6]. This system is capable of reasoning upon accumulated knowledge, as well as the knowledge acquired while processing text, to test hypotheses and define semantics of words and phrases more precisely. The hypotheses that were confirmed extend SOFIE's ontology, and the system receives new extraction templates (linguistic rules), which further can be used for information extraction.

This paper presents a system that implements an ontological approach to extracting facts from text. Unlike existing systems, which target mainly extraction of the desired category of facts mentioned explicitly in the text of the analyzed document, this system allows obtaining new facts not stated in the text explicitly. The proposed approach improves the completeness of fact extraction due to: (1) use of ontologies to extract facts from the text and (2) deriving the facts not mentioned in the texts. The inference of implicit facts is based on the facts acquired in text analysis and the basic knowledge of the ontology.

3 The Approach and Basic Models

Under a *thematic fact* (TF) we understand an assertion characterizing some entity S (the subject of the fact) in a certain aspect. The aspect, in which the subject of the fact is characterized, defines the base relation R connecting S to another entity O (the object of the fact). Thus, a thematic fact can be formally represented using the language of binary relations:

$$TF = R(S, O) \ .$$

The relation R specifies the corresponding *fact category* (FC).

The proposed approach is implemented in the context of developing an IAS designed for evaluation of innovative technologies. For that reason new technologies are considered as subjects of the facts. However, the proposed approach is universal and can be used in other areas.

Particular aspects characterizing technology (generally, the fact subject) determine the appropriate categories of facts to be extracted from unstructured texts. Such categories include, for instance, companies that develop the technology, the readiness degree of the technology, companies that are potential consumers of the technology etc. The list of fact categories is known in advance and is used in the system design.

Two main issues impede the fact extraction process:

1. Skipping some relevant facts contained in texts due to the variety and complexity of their possible formulations. Facts can be not retrieved due to:
   - *Lexical* diversity of TF's structural elements: subjects, objects and relations. An example of the base relation synonymy: "Enterprise E is *developing* technology T" and "Enterprise E is *working on creation* of technology T".
   - *Syntactic* diversity of TF's expression. The order of TF's structural elements in a sentence is not strictly fixed, for instance: "Enterprise E is developing technology T" and "Technology T is being developed by enterprise E".
2. The absence of explicit mentions of relevant facts in texts, despite the fact that such facts can be logically inferred from those already found in texts using some general knowledge.

The proposed approach addresses these problems and provides an increase of the fact extraction completeness, which is interesting for the system's user (an expert). Specific aspects of the approach are discussed in detail below.

3.1 Using Ontologies for Thematic Fact Extraction

Facts of various categories are extracted from texts with the use of corresponding *extraction template* (ET). To improve the completeness of TF extraction, elements of templates that correspond to different FCs are associated with elements of a *lexical ontology*. This allows to fully use the possible lexical expression forms of TF subject, object and relation, as well as clearly define their semantic identity. For each given FC a set of ETs was developed. These templates determine generically the category-specific subject, object and relation as elements in lexical ontology. For example, the ET for a FC about the companies-consumers of a given technology can be generally defined as follows:

$$element\_specifying\_Interest(element\_specifying\_Technology, element\_specifying\_Enterprise) \quad (1)$$

The problem of syntactic diversity of a TF is solved by ontological description of the ET structure by listing its elements without strictly fixing their order.

The OBIE approach enables one to use a lexical ontology for managing the fact extraction process. To organize this, all developed ETs are specified in the ontology using the corresponding *template relations*. Template relations are binary relations linking the entity, representing the FC template, with entities, representing each of the template elements. There exist three relations of this kind for each ET:

$$is\_Template\_Subject(FC\_template, ET\_subject),$$
$$is\_Template\_Object(FC\_template, ET\_object),$$
$$is\_Template\_Relation(FC\_template, ET\_relation).$$

Thus the lexical ontology defines:

1. Entities corresponding to the elements of ETs for each FC;
2. Hierarchical relations and relations of lexical synonymy;
3. Template relations for every ET defined in the IAS.

The entities of high level abstraction in the lexical ontology include the following: Fact_Category (FC), Fact_Subject (FS), Fact_Relation (FR) and Fact_Object (FO). When developing ET the concrete subclasses of these high-level classes are defined. These subclasses then specify corresponding template relations among each other. The elements of ETs are represented by *descriptive instances* of ontological classes matching the defined relation, its subject and object. A descriptive instance (DI) is an instance of an ontological class which implements its object properties. DIs allow to formalize ontological knowledge. They specify a set of characteristic object relations, their domains and ranges at the class level, and allow to implement these relationships for specific instances. For every DI a set of *lexical instances*, belonging to the same class, can be defined. A lexical instance (LI) is an instance of an ontology class, which expresses the lexical word form of the class. All the LIs specified for a DI are semantically identical, i.e. they are lexical synonyms.

Let us consider an example demonstrating the concepts of DI and LI. There exists a generic class named "Fact_Category", which has a relation "has_Relation" with the range of class "Fact_Relation". We will consider the FC of enterprises, which are consuming some technology; the corresponding ET is specified as (1). Based on (1), a subclass named "Enterprises_Consumers" is created for the "Fact_Category" class, and for the "Fact_Relation" class a subclass named "Interest" is created. It is not possible to connect the new classes with the "has-Relation" relation directly, and for each of them corresponding DIs are created, which are linked by this relation. Then, for the "Enterprises_Consumers" and "Interest" classes lexical filling is to be specified, i.e. LIs are to be created. To connect a set of LIs to an ET, there exists a "hasLexicalForm" relation, which connects every LI to the corresponding DI of its class. For instance, the "Interest" class may contain LIs named "interested", "plans to buy", etc.

Having one or more LIs for all structural components of the template (in a random order) in the structural element of a document is the basis for the extraction of this element as a thematic fact.

Fig. 1 shows a fragment of the lexical ontology that describes knowledge about enterprises-consumers of the required technology. Considering the introduced definitions, the ET for this category is specified as follows:

$$DI\_of\_Interest\_Class(DI\_of\_Technology\_Class, DI\_of\_EnterprisesConsumers\_Class)$$

On Fig. 1 the elements of the ontology are shown in the following way: high-level classes and basic subclasses participating in fact extraction process – in thick borders; their instances – in dotted-line borders; the subclass relation ("isSubclassOf") – (*); the type relation ("isA") – dotted-line arrows; the lexical form relation ("hasLexicalForm") – (**). Number "1" specifies the template's relation "hasObject", number "2" – the template's relation "hasRelation", number "3" – the template's relation "hasSubject".

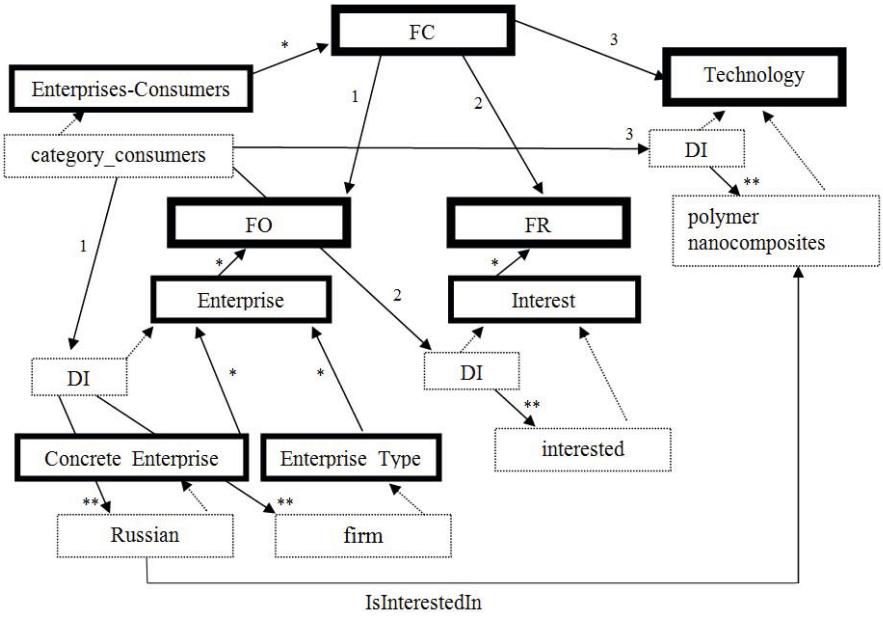

Fig. 1. A fragment of the lexical ontology describing knowledge about enterprises-consumers of the required technology

The lexical ontology expands during the text analysis process by the *extracted TF* (EF). Each EF is normalized and added to the ontology as a statement (a triple) represented by an *object property* specified for the considered FC. This object property links the lexical forms of the TF subject and relation, which

were found in texts. For instance, for the considered FC the lexical ontology may expand by a triple "IsInterestedIn (polymer nanocomposites, Russian Helicopters)". This new relation is shown in Fig. 1.

3.2 Deriving Facts Based on the Basic Knowledge and Facts Extracted in the Document Analysis Process

The discussion of the approach to deriving of relevant facts not contained explicitly in texts (derived facts, DF) we begin with an example. Let us assume that an expert is interested in facts about the companies being potential consumers of some technology T. We also assume that for certain classes of products, there exists a basic knowledge (i.e. knowledge specific to a given area) of two types stored in the ontological knowledge base:

- Knowledge about the structure of the products as a hierarchy of items that make up the given class of products: subsystems, components, elements; subproperties of "hasPart" property (e.g. "hasSubsystem", "hasComponent") and its inverse property "partOf" are used;
- Knowledge about the materials used in the manufacturing of certain subsystems (components). Here the "usedMaterial" property is used (and its inverse property "isUsedIn").

Let us suppose that in the process of text analysis a fact was found stating that the analyzed technology T is perspective for the production of material M. In addition, another fact has been extracted stating a certain company is planning to produce a certain type of product P. Using the basic knowledge that this type of products contain components which use material M in their manufacturing it can be concluded that the enterprise is a potential consumer of technology T. Here is a possible inference rule for this kind of facts:

$$\begin{aligned} &hasSubsystem(?Product, ?Subsystem) \wedge \\ &hasComponent(?Subsystem, ?Component) \wedge \\ &isUsedIn(?Material, ?Component) \wedge \\ &isUsedFor(?Technology, ?Material) \wedge \\ &planToProduce(?Enterprise, ?Product) \\ &\Longrightarrow \\ &isPotentialConsumer(?Technology, ?Enterprise) \end{aligned} \tag{2}$$

The model of deriving of implicit facts can be formally written as follows:

$$(BK, EF) \underset{IR}{\models} DF$$

where BK – *Background Domain Knowledge*; EF – *Extracted Facts*; IR – *Inference Rules*; DF – *Derived Facts*.

The algorithms of system's prototype operation that implements this approach to facts extraction are presented in the next section.

4 Algorithms

This section presents two algorithms which implement the approach discussed above: i) extracting TF that are explicitly stated in documents and ii) deriving of facts using basic knowledge and facts, extracted from texts.

Let us introduce the required abbreviations: *LexOntology* — lexical ontology; *query* — user query; {*FC*} — the array of thematic fact's categories; {*IFC*} — the array of inferred facts categories; *FS* — fact subject; *FO* — fact object; *FR* — fact relation; *DC* — document corpora that is being processed; *Doc* — a document; Doc_{pt} — a document in plain text; *DI* — descriptive instance; {*fact_category_tag*} — the array of fact categories tags; {*IR*} — inference rules; {*extracted_fact*} — the array of extracted facts; {*inferred_fact*} — the array of logically inferred facts; *BackgroundKnowledge* — basic domain knowledge. The process of extracting facts from documents is presented in Algorithm 1.

On step 1 the *descriptive instance* (DI), corresponding to the FS is extracted from the lexical ontology. On step 2 the subject of the current query is added to the ontology using the *hasLexicalForm* relation of the extracted DI. In cycle 3 all of the documents of the corpora being analyzed are converted to plain-text (4), divided into sentences (5), and the sentenced that are considered to be potential facts (according to the user query) are detected (6). Step 7 introduces a cycle on all the potential facts; this cycle contains another cycle on fact categories (8).

Step 9 extracts the DI for the ontological class representing the current FC. On step 10 following the *hasObject* relation the DI of the class representing the FO for the FC is extracted. On step 11 the DI of the class representing the relation of the FC is extracted using the *hasRelation* relation.

On steps 12-13 all the lexical forms for FO and FR DIs are extracted from the lexical ontology by the *hasLexicalForm* relation. Step 14 forms an extraction template from the lexical forms, and the extraction template is matched against the potential fact in step 15. If a match was found, step 17 extracts an instance from the lexical ontology which is corresponding to the lexical form of the FO. Step 18 extracts a relation which is specific for the current FC. Finally, on step 19 the extracted TF is added to the lexical ontology. The sentence under consideration is tagged with the FC tag (20) and is added to the extracted facts array on step 21.

For each of the inferred fact categories a cycle 1 is organized: on step 2 an inference rule specific for the IFC is determined. On step 3, using the metadata stated for all the predicates in the inference rule, the array of facts to be extracted is formed. On step 3, using the metadata stated for all the predicated in the extraction rule, the basic knowledge to be used to derive new facts is formed. Steps 7-8 address the Algorithm 1, which allows to extract the required facts from the initial document corpora. The extraction template is already specified for this stage: the object and the relation of the pattern are determined by their lexical forms. If the required fact has been extracted, a new fact is derived using the basic knowledge (step 10). On step 12 the derived fact is added to the array of derived facts.

Algorithm 1. Extracting thematic facts from texts

input $LexOntology$, $query$, $\{FC\}$, $\{fact\_category\_tag\}$, DC
output $\{extracted\_fact\}$, $LexOntology$
$FSDI \leftarrow getDescInstance(LexOntology)$
$User\_Query\_Instance$
$\leftarrow updateLexOnto(hasLexicalForm(FSDI, query), LexOntology)$
for all $Doc \in DC$ **do**
  $Doc_{pt} \leftarrow toPlainText(Doc)$
  $\{Sentence\} \leftarrow splitText(Doc_{pt})$
  $\{PossibleFactSentence\} \leftarrow getPossibleFacts(FS(query), \{Sentence\})$
  for all $PossibleFact \in \{PossibleFact\}$ **do**
    for all $fc \in \{FC\}$ **do**
      $FCDI \leftarrow getDIForFactCategory(LexOntology, fc)$
      $FODI \leftarrow getDIForFactObject(LexOntology, hasObject(FCDI))$
      $FRDI \leftarrow getDIForFactRelation(LexOntology, hasRelation(FCDI))$
      $\{FO\_lexical\_form\} \leftarrow getLexicalForms(LexOntology,$
        $hasLexicalForm(FODI))$
      $\{FR\_lexical\_form\} \leftarrow getLexicalForms(LexOntology,$
        $hasLexicalForm(FRDI))$
      $lexicalPattern \leftarrow formLexicalPattern(\{FO\_lexical\_form\},$
        $\{FR\_lexical\_form\})$
      $[is\_Match, MatchedSubjectLexicalForm]$
        $\leftarrow conductPatternMatching(PossibleFact, lexicalPattern)$
      if is_Match **then**
        $FOLexicalFormInstance \leftarrow getInstance(LexOntology,$
          $MatchedSubjectLexicalForm)$
        $Relation \leftarrow getOntologicalRelationForFC(LexOntology, fc)$
        $LexOntology \leftarrow updateOntology(LexOntology,$
          $FOLexicalFormInstance, User\_Query\_Instance, Relation)$
        $Fact \leftarrow tagFact(tag, PossibleFact)$
        $\{extracted\_fact\} \leftarrow addFact(\{extracted\_fact\}, Fact)$
      end if
    end for
  end for
end for

5 Implementation

The proposed approach to facts extraction has been implemented in the FactE framework. FactE is a subsystem of Information-analytical system which takes the user query containing the name of the innovative technology, as well as a corpus of texts to be processed, as an input. The list of fact categories and inference rules for deriving of new facts are believed to be given in advance.

The FactE architecture has been developed on the basis of the above presented algorithms. As seen from Fig. 2, FactE has two functional modules:

Algorithm 2. Deriving of facts using basic knowledge and texts

input $\{IFC\}$, $\{IR\}$, BackgroundKnowledge
output $\{inferred\_fact\}$
1: **for all** $ifc \in \{IFC\}$ **do**
2: $IR \leftarrow getInferenceRuleForCategory(ifc, \{IR\})$
3: $\{Required\_Fact\} \leftarrow getRequiredFacts(IR)$
4: $\{Required\_Fact\} \leftarrow getRequiredFacts(IR)$
5: $\{Background\_Knowledge\} \leftarrow getBackgroundKnowledge(IR,$ $BackgroundKnowledge)$
6: **for all** $required\_fact \in \{Required\_Fact\}$ **do**
7: **for all** $Doc \in \{DC\}$ **do**
8: Apply **Algorithm 1**, steps [2, 4-5, 6-8, 14-15] with input: $FC = ifc$, $FO = required\_Fact.Object$, $FR = required\_Fact.Relation$
9: **if** is_Match **then**
10: $Inferred\_Fact \leftarrow applyRule(PossibleFact,$ $\{Background\_Knowledge\})$
11: **end if**
12: $\{Inferred\_Fact\} \leftarrow addInferredFact(\{Inferred\_Fact\},$ $Inferred\_Fact)$
13: **end for**
14: **end for**
15: **end for**

- linguistic processor, designed for fact extraction using filled extraction templates, which are provided by the lexical ontology and
- fact derivation module, which is intended to organize the logical inference of new facts.

The functioning of the mentioned modules is controlled by the system ontology which describes and stores both lexical knowledge (managing the process of extraction of implicitly mentioned facts) and basic knowledge (used for new facts inference).

The following third-party software was used for solving particular problems when implementing the FactE framework: Apache HttpClient [7], Apache Tika [8], GATE [9], Apache Open NLP SentenceSplitter [10], Apache Lucene [11], Apache Jena Core [12], Apache Jena SDB [13].

6 Preliminary Experimental Results

The system's prototype is now at the early development stage. Particularly, the amount of data in domain ontologies containing basic knowledge is small, and the set of the experimental results obtained is limited at this point. Below a real result obtained while testing the prototype is described. The testing was held on the basis on Internet sources in Russian language.

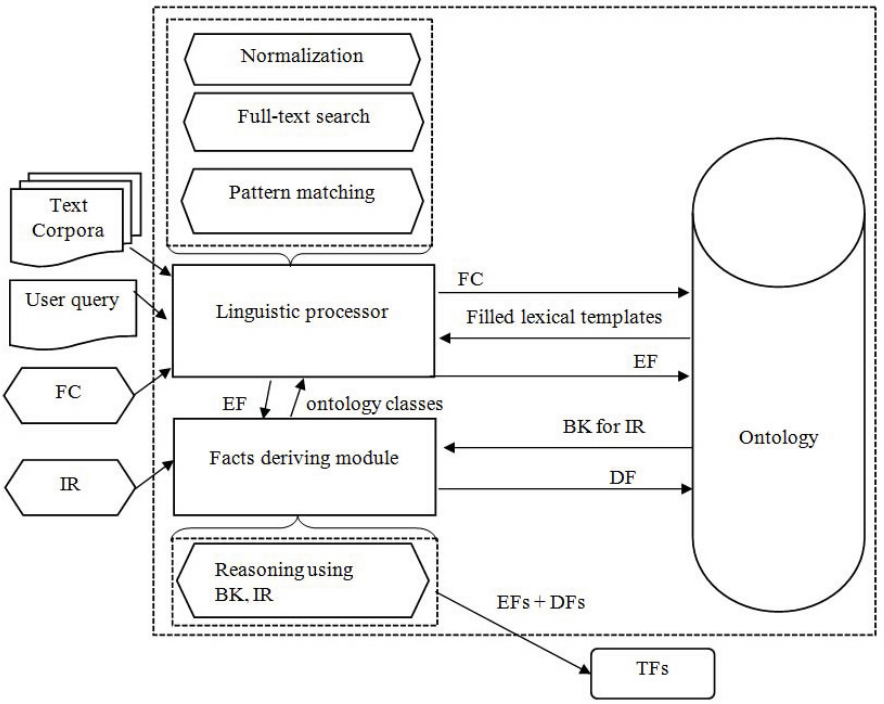

Fig. 2. FactE architecture

The query contained the technology of producing polymer composites. The fact category was "potential consumers of the technology". In the analysed text corpora the document [14] was included amongst others. This document contains the sentence "The joint company "Helicopters of Russia" intends to create a perspective multipurpose commercial helicopter". The algorithm of fact extraction reveals the following fact:

$$Plans\_To\_Produce(\ll Helicopters\ of\ \ Russia \gg, helicopter) \tag{3}$$

The background knowledge of FactE includes the following statements:

$$Includes(helicopter, ballscrew) \tag{4}$$

$$Contains(ballscrew, blades) \tag{5}$$

$$Used\_for\_Production(CPRF, blades) \tag{6}$$

$$May\_Be\_Used\_For\_Manufacturing\_Material(polymernanocomposites, CPRF) \tag{7}$$

As a result on the basis of the explicit fact (3), background knowledge (4)–(7) and inference rule (2) the following fact of the category "Enterprises-Potential consumers of technology T" have been inferred by FactE:

$$Is\_Potential\_Consumer(\ll Helicopters\ of\ Russia \gg, polymernanocomposites)\ . \quad (8)$$

where T stands for the given technology (that is, polymer nano-composites).

7 Conclusion and Further Work

The paper discussed an approach to thematic facts extraction, allowing to fetch explicitly stated facts and derive new facts using the domain knowledge and already extracted facts. The FactE framework implementing the proposed approach was presented, its architecture and operational algorithm were discussed. Future works include improving the quality of text analysis by using more complex linguistic tools which would allow to solve the problem of coreference and context resolution. Above all, a consistency check for the ontology is to be introduced to help avoid addition of the statements which are known to be wrong in the given domain.

References

1. Feldman, R., Sanger, J.: The Text Mining Textbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge Univ. Press (2007)
2. Wimalasuriya, D., Dou, D.: Ontology-based information extraction: An introduction and a survey of current approaches. J. of Inf. Science 36(3), 306–323 (2010)
3. Anantharangachar, R., Ramani, S., Rajagopalan, S.: Ontology Guided Information Extraction from Unstructured Text. Int. J. of Web & Sem. Tech. 4(1), 19–36 (2013)
4. Buitelaar, P., Cimiano, P., Frank, A., Hartung, M., Racioppa, S.: Ontology-based Information Extraction and Integration from Heterogeneous Data Sources. Int. J. of Human Computer Studies 66, 759–788 (2008)
5. Petasis, G., Möller, R., Karkaletsis, V.: BOEMIE: Reasoning-based Information Extraction. In: Proceedings of the 1st Workshop on Natural Language Processing and Automated Reasoning, pp. 60–75 (2013)
6. Suchanek, F.M., Sozio, M., Weikum, G.: SOFIE: A self-organizing framework for information extraction. In: Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, pp. 631–640 (2009)
7. Apache Http Client, `http://hc.apache.org`
8. Apache Tika, `http://tika.apache.org/`
9. GATE: General Architecture for Text Engineering, `https://gate.ac.uk/`
10. Apache Open NLP, `https://opennlp.apache.org/`
11. Apache Lucene, `http://lucene.apache.org`
12. Apache Jena Core, `https://jena.apache.org/documentation/rdf/`
13. Apache Jena SDB, `http://jena.apache.org/documentation/sdb/`
14. A PROMISING HIGH-SPEED HELICOPTER (PSV) V-37, `http://bastion-karpenko.ru/v-37_psv/`