Lang Resources & Evaluation
DOI 10.1007/s10579-017-9387-6

ORIGINAL PAPER

# A flexible text analyzer based on ontologies: an application for detecting discriminatory language

**Alberto Salguero**[1] · **Macarena Espinilla**[2]



**Abstract** Language can be a tool to marginalize certain groups due to the fact that it may reflect a negative mentality caused by mental barriers or historical delays. In order to prevent misuse of language, several agents have carried out campaigns against discriminatory language, criticizing the use of some terms and phrases. However, there is an important gap in detecting discriminatory text in documents because language is very flexible and, usually, contains hidden features or relations. Furthermore, the adaptation of approaches and methodologies proposed in the literature for text analysis is complex due to the fact that these proposals are too rigid to be adapted to different purposes for which they were intended. The main novelty of the methodology is the use of ontologies to implement the rules that are used by the developed text analyzer, providing a great flexibility for the development of text analyzers and exploiting the ability to infer knowledge of the ontologies. A set of rules for detecting discriminatory language relevant to gender and people with disabilities is also presented in order to show how to extend the functionality of the text analyzer to different discriminatory text areas.

**Keywords** Text analyzer · Document text model · Methodology · Ontology · Discriminatory language

✉ Alberto Salguero
alberto.salguero@uca.es

Macarena Espinilla
macarena.espinilla@ujaen.es

[1] Department of Computer Sciences, University of Cádiz, Cádiz, Spain

[2] Department of Computer Sciences, University of Jaén, Jaén, Spain

Published online: 05 April 2017

## 1 Introduction

Communication is an essential aspect among humans because we are naturally sociable and we need to express our feelings and emotions. Thus, the language used by people is the translation of their own thoughts and realities, i.e, it provides our own interpretation of the world (Dance 1970).

Sometimes, unintentionally, the language reflects a negative mentality caused by mental barriers or historical delays. As Gadamer stressed *language is a carrier of traditions and prejudices* (Tontti 2004). Therefore, the language can be used as a tool to marginalize certain groups of people. However, the inequality in the language should be managed in the society because people should be treated with dignity and respect (Claude and Weston 1992), starting the pursuit of equality in the language with education on the proper use of terms and phrases (Ahmed 2007; Augoustinos et al. 2005; Yates 2001).

Governments in many countries have enacted laws to correct the imbalances in the language which are responsible for a situation of discrimination. Similarly, multiple agents such as institutions, unions or nongovernmental foundations have carried out campaigns against discriminatory language, criticizing the use of some terms and phrases (Colker and Milani 2012; Mowbray 2012; Loenen and Rodrigues 1999; Schiek and Lawson 2011; Shuy 2007). As a result, manuals and guides have been developed that classify discriminatory language by different conditions such as gender, race, ethnicity, sexual orientation, religion, age or disability (Brading and Curtis 2000; Kubota and Lin 2010; Orelus 2011; Weller et al. 2013).

The most studied condition of discrimination is gender or sexist language that promotes and encourages gender discrimination against women. This is the most analyzed and discussed language because it is the language that has more characteristics, has several variants and it is the most widespread marginalized group (Talbot 2010; Litosseliti 2014). In this context, a marginal practice is for example the use of *he* or *his* when referring to both a *female* and a *male*, excluding the female. To be inclusive, both *he* and *she* pronouns should be used and these must consciously balance pronoun use. For example, *"If a student studies hard, he will succeed"* is a marginal sentence while *"If a student studies hard, he or she will succeed"* is an inclusive sentence. Another example in the context of disability is the use of the term *paraplegics*, being the suggested alternative *"people with paraplegia"*. Those kinds of expressions as well as other potential discriminatory expressions are considered incidences and must be avoided.

Despite the importance of combating discriminatory language in the last two decades, there is a gap in the proposed approaches to detect discriminatory text in the written language. In the literature we can find that in Chen et al. (2012) a lexical syntactic feature architecture was proposed to detect offensive content, introducing syntactic rules. In Kontostathis et al. (2009) a rule-based communication model to track and categorize online predators using a history of conversations was presented. In order to detect vandalism in Wikipedia, in Alfonseca et al. (2013) an approach was proposed based on the revision history and in Chin et al. (2010) an active learning approach that uses language model statistics was presented.

Notwithstanding the usefulness of the proposed approaches in their own fields of application, these approaches present two important barriers. The first barrier is that the language is very flexible and there are some hidden features of words or relationships among them. For this reason, some proposals are less efficient when these proposals try to detect hidden problems or incidences. The second barrier is that, usually, these proposals are focused on one type of incidence or purpose for a specific field. The possibility to include the detection of another type of incidence or modify the purpose of the text analyzers is very complex because the developed proposal is strongly linked to its aim.

In this paper, we are focused on the analysis of text documents that implies the discrimination of marginal groups. To provide a solution to overcome the first barrier, we propose an ontology-based text model as well as a methodology to develop a basic text analyzer based on the proposed model for discriminatory language detection. We take the advantages of reasoning mechanisms of ontologies to detect patterns among the entities in the document without the need of explicitly recording all the relationships among them. So, in this paper, ontologies are used to represent the text document, including concepts such as *Term*, *Sentence* or *Noun*.

With the aim to provide a solution to overcome the second barrier, which is related to the lack of flexibility, the proposed model has been designed to be extensible, so it can also be used to detect other kinds of discriminative problems or completely modify its purpose. To increase the functionality of the basic text analyzer, sets of new rules and concepts can be added to the basic ontology-based document text model. Those rules can be used in the proposed methodology to detect when a word or sentence is an incidence of a discriminatory problem.

Therefore, the main aim of our proposal is based on the use of ontologies to build a text analyzer with a high level of flexibility in order to overcome the previous two barriers. Furthermore, in order to illustrate the simplicity and usefulness of the proposed model, a set of rules for detecting discriminatory language is presented. We present some common rules for the detection of general discriminatory expressions as well as some of the rules to detect discriminatory language regarding to gender and people with disabilities.

The paper is structured as follows: the next section provides a revision about related research in the fields of ontologies and text analyzers. The third section introduces the model to represent text documents using ontologies. The methodology to develop a text analyzer using this model is proposed in the fourth section. The fifth section is dedicated to explain the concepts and rules that extend the methodology for detecting some types of discriminatory language. Finally, conclusions and future works are pointed out in the last section.

## 2 Ontologies and text analyzers

In this section, we provide a brief review about ontologies and some related concepts and tools that will be used in our proposals. Furthermore, we revise some text models in the literature that are based on ontologies in order to provide a brief revision of the state of the art in the study field.

### 2.1 Introduction to ontologies and the Ontology Web Language (OWL)

Ontologies are used to provide structured vocabularies that explain the relations among terms, allowing an unambiguous interpretation of their meaning. So, ontologies are formed by concepts (or classes) which are, usually, organized in hierarchies (Chandrasekaran et al. 1999; Uschold and Gruninger 1996), being the ontologies more complex than taxonomies because they not only consider *type-of* relations, but they also consider other relations, including *part-of* or domain-specific relations (Knijff et al. 2013).

In an ontology, the symbol $\top$ stands for the *top* concept of the hierarchy, all being concepts subsets of $\top$. The *subsumption* relation is usually expressed using the symbol $A \sqsubseteq B$, meaning that the concept $A$ is a subset of the concept $B$. Concepts can also be specified as logical combinations of other concepts.

An ontology expresses what individuals, also called objects, belong to which concepts. Moreover, it is possible to declare properties to relate individuals, organizing them into a hierarchy of sub-properties, and providing domains and ranges for them. Usually, the domains of properties are concepts and ranges are either concepts or data types. A declared property can be defined as transitive, symmetric, functional, or the inverse of another property ($A^{-}$).

The main advantage of ontologies is that they codify knowledge and make it reusable by people, databases, and applications that need to share information (Knijff et al. 2013; Wei et al. 2015). Due to this, the construction, the integration and the evolution of ontologies have been critical for the so-called Semantic Web (Horrocks 2008; Kohler et al. 2006; Maedche and Staab 2001). However, obtaining a high quality ontology largely depends on the availability of well-defined semantics and powerful reasoning tools.

Regarding Semantic Web, a formal language is OWL (Horrocks et al. 2003; Sirin et al. 2007), which is developed by the World Wide Web Consortium (W3C). Originally, OWL was designed to represent information about categories of objects and how they are related. OWL inherits characteristics from several representation languages families, including the Description Logics and Frames basically. OWL is built on top of the Resource Description Framework (RDF) and RDF Schema (RDFS). RDF is a datamodel for describing resources and relations between them. RDFS describes how to use RDF to describe application and domain specific vocabularies. It extends the definition for some of the elements of RDF to allow the typing of properties (domain and range) and the creation of subconcepts and subproperties. The major extension over RDFS is that OWL has the ability to impose restrictions on properties for certain classes.

The design of OWL is greatly influenced by DL, particularly in the formalism of semantics, the choice of language constructs and the integration of data types and data values. In fact, OWL DL and OWL Lite (subsets of OWL) are seen as expressive DL, offering a DL knowledge base equivalent ontology.

One of the main advantages of the high formalization of OWL is the possibility of using automated reasoning techniques. In 2009, the W3C proposed the OWL 2 recommendation in order to solve some usability problems detected in the previous version, keeping the base of OWL. So, OWL 2 adds several new features to OWL,

some of the new features are syntactic sugar (e.g., disjoint union of classes) while others offer new expressivity, including: increased expressive power for properties, simple metamodeling capabilities, extended support for datatypes, extended annotation capabilities, and other innovations and minor features (Zhang et al. 2015).

Another key tool that is usually used in conjunction with OWL is SPARQL,[1] which is a query language able to retrieve and manipulate data stored in RDF triples. The main difference with SWRL is that the SPARQL language has been designed to work with RDF triples, at a lower abstraction level. SWRL has been built on top of OWL, extending the set of its axioms, whereas SPARQL is designed to work with individuals. Therefore, it is mainly used to retrieve or modify individuals meeting certain conditions.

OWL is heavily based on formal semantics, so there are some situations in which SPARQL is extremely useful. It overcomes some of the limitations of OWL when Open World Assumption (OWA) issues arise. OWA is the assumption that what is not known to be true is simply unknown, not false. A term that is not annotated as *Singular* does not have to be *Plural* in OWL, for example. On the contrary, it is possible to use SPARQL to get all those terms that have not been annotated as *Singular*. Furthermore, it also supports aggregations, which are very useful for the extraction of information from ontologies.

### 2.2 Ontology-based text analyzers

Traditional text classifiers usually rely on the Bag of Words approach, which ignores the structure of texts. Thus, the contextual information of words is lost. In the literature, we can find that many authors have tried to combine general-purpose thesauri as Wordnet or Wikipedia and domain specific ontologies in order to analyze textual data.

In Li et al. (2012) some algorithms are reviewed to analyze texts and classify them, using WordNet to improve the results. To do so, the accuracy of classifiers are increased by considering the similarity among terms as a feature for the supervised learning algorithms. The method to compute the similarity between two concepts takes into account the shortest path length between two terms and the depth of the subsumer in the WordNet hierarchy (Hotho et al. 2002).

The approach proposed in Wei et al. (2015) exploits ontology hierarchical structure and relations to provide a more accurate assessment of the similarity between terms for word sense disambiguation.

WordNet is also used in Luo et al. (2011) to annotate the terms in texts as well as categories containing a set of related concepts that are closely related to synsets, the sets of synonyms in WordNet. A term weighting scheme is proposed in which the weight of each term is dependent on its semantic similarity to the category.

The authors of the proposal presented in Kontopoulos et al. (2013) use WordNet to analyze and classify Twitter posts. Categories are assigned to the topic at hand with the help of a semi-automatic tool. This tool offers assistance during the development of domain ontologies, by suggesting concepts and relations, and creating instances of the

[1] http://www.w3.org/TR/sparql11-query.

concepts in an automatic way. The developed ontology is, in essence, a simple taxonomy of concepts and attributes, which is enriched with synonyms and hyponyms relations that are used to improve the accuracy of the classifier.

Most of these proposals use WordNet as an ontology rather than as a mere lexical resource. In most cases, these approaches only make use of synonymy relation defined between synsets in WordNet. Indeed, WordNet contains an excellent coverage of both the lexical and conceptual palettes of the English language. However, WordNet has not been formally axiomatized so as to make the logical relations among the concepts precise. In spite of this disadvantage, WordNet is still used as an ontology due to the fact that some of its lexical links are interpreted according to predefined formal semantics (Gangemi et al. 2003). In the literature, we can find some proposals that work in the same way, using lexical resources such as Wikipedia or Open Directory (Gabrilovich and Markovitch 2007; Wang et al. 2009).

Some domain specific ontologies have been proposed to analyze texts. In Machhour and Kassou (2013) a method to classify text documents based on predefined ontologies was proposed. Given a document, an algorithm determines the most relevant concepts in a domain ontology. To do so, synonym relations established among concepts in the ontology as well as a list of related concepts are used to classify the document using some supervised learning techniques.

The proposal presented in Garla and Brandt (2012) also uses the similarity among concepts to classify clinical documents. In addition to the synonym relation, the hypernym relation is also defined in a clinical ontology. The proposal is based on the fact that if a concept is not relevant to a classification task, then the similarity among its concept's descendants is also not relevant.

Mimir (Tablan et al. 2015), a semantic search framework, uses automated reasoning to improve the quality of text analysis. Based on Ontotext,[2] Mimir uses linguistic annotations created by GATE,[3] document structure annotations and Open Linked Data to create an index that complements information seeking searches. For example, a query on flooding in the United Kingdom would retrieve a document about floods in Cambridge, even though the latter does not explicitly mention the UK.

NLTK (Bird et al. 2009) allows the translation of the meaning of texts into first-order logic expressions, on the basis of a syntactic parse, for carrying out automated inference. The result can be used to test facts about the discourse such as "Every student is a person".

IKS is an open source community, whose projects are focused on building a platform for semantically enhanced Web Content Management Systems. BESA-HOT (Kasper and Vela 2012), one of those projects, is a service that collects user reviews for hotels from various sites on the Web, analyzes them and classifies the textual content of the reviews according to their sentiment polarity. The system not only extracts the relevant textual content of the review but also other metadata such as scores and information about the reviewer. All the extracted review content,

[2] http://ontotext.com/.

[3] https://gate.ac.uk.

including metadata, is represented as a RDF instance of a review ontology defined in OWL. A rule system is then proposed to supply answers to questions such as what is evaluated (topic) and what properties are evaluated (dimension).

Regular expressions are also used in Aussenac-Gilles and Sörgel (2005) to identify concepts and their relations in texts. Patterns are formed with words, jokers, POS, words characterizations or semantic classes. The text is analyzed and, when some of the patterns is matched, the user is prompted to validate it. In such case, individuals and relations among them are created according to the pattern. This approach assumes that linguistic regularities may reveal similar knowledge and, particularly, similar semantic relations (Hearst 1992).

In Buitelaar et al. (2004) the regular expressions are specified using XPath and they make use of the syntactic structure of texts. In this approach, texts are previously annotated by adding tags in XML. To do so, an automatic tool is used that is able to automatically annotate words with POS information given its context. Rules may be defined for matching all head nouns that are relevant for a specific domain, for instance.

The Simple Knowledge Organization System[4] (SKOS) is used in Machhour and Kassou (2013) to represent the terms in the texts. SKOS is a general vocabulary designed for representing thesauri, taxonomies and other classification schemes. Because the entities are described in RDF, the information is exchanged easily among applications (Isaac and Summers 2009).

Lemon (McCrae et al. 2012) is a RDF model for modeling lexicon and machine-readable dictionaries. The Lemon core module is intended to have a similar expressive power to that of SKOS, while providing distinctions that allow for more powerful linguistic modeling. Although it is focused on the sense annotation of lexical entries, its phrase structure module allows the representation of sentences as lists of words to some degree. This is done through the usage of RDF lists of component objects. Lists are recursively defined as pairs containing a head element and a list of items. This representation is too verbose and makes information associated to the structure of the sentences difficult to be exploited by reasoners. It is very difficult to determine if a word is preceded by given word, for example. Actually, this mechanism is just intended for the representation of multi-word lexical entries. In any case, it is not possible to represent a whole text document in the Lemon model.

The Ontolex model (Cimiano et al. 2016) is based on the Lemon model (McCrae et al. 2012). Its purpose is to support linguistic grounding of a given ontology by adding information about how the elements in the vocabulary of a given ontology are lexicalized in a language. Beyond the *constituent* property, which can be used to specify the components of a phrase (multiword expressions, mainly), the model does not provide resources for describing the structure of texts.

The idea behind NIF (Hellmann et al. 2013) is to allow NLP tools to exchange annotations about texts in RDF. The NIF Core Ontology provides classes and properties to describe the relations between substrings, texts and documents by assigning URIs to strings. These URIs can then be used as subjects in RDF triples

[4] https://www.w3.org/2004/02/skos/.

and therefore they can be annotated easily. The core of the model includes a lot of classes and properties to describe the structure of text documents, such as *Title*, *Context* (source) or *Paragraph*, but most of them are useless in our case. Being the text model that is more similar to our proposal, the verbosity of NIF prevents its use in other research areas such as DL Class Expression Learning, one of the main objectives for the development of the proposed model. The formalization of the list pattern allows us to re-use it for the representation of both list of sentences and list of words, instead of duplicating them as the NIF model does, for example.

After this brief review, we can see that the ontologies work with flexible and generic data models and can be easily extended and combined with other ontologies. However, previous related research works which are focused on text analysis have often taken into account explicit features but ignored implicit features (Xu et al. 2015). Only some of them uses inferred knowledge, which is one of the main advantages of ontologies. The knowledge contained in the ontologies could be used to improve the quality of text analysis tools, although little effort has been made to exploit the full reasoning capabilities of ontologies.

## 3 Flexible ontology-based text model

In this paper, an ontology to formally define concepts that describe basic elements of text processing is proposed. This ontology is used to represent text in a machine readable form, so automatic reasoning can be applied to extract knowledge. This allows us to extract information from the text document by using logical rules. To do so, in this section, we first introduce the structure of the ontology-based text model, which is based on a list pattern. Then, some of the concepts that will be used to describe the entities in the documents are presented. The ontology has been created following the Uschold and Gruninger methodology for building ontologies (Uschold et al. 1996).

### 3.1 List pattern

In order to identify the entities in the text and the relations among them, our proposal for representing text documents is based on a list structure. The underlying RDF collections (Hayes and Patel-Schneide 2014) are unavailable because they are used in the RDF serialization of OWL. Although *rdf:Seq* is not illegal, it depends on lexical ordering and has no logical semantics accessible to a DL classifier (Drummond et al. 2006).

In the literature, we can find some proposals for the representation of lists in OWL. In ODP (2010) a design pattern for representing them is proposed. However, because the concept *Collection* is defined to be disjoint with the concept *item*, it is not possible to represent lists having other lists as items. This is the case of our representation, where the text is represented as a sequence of sentences which, in turn, are represented as sequences of words. Two general design patterns for representing lists are also proposed in Drummond et al. (2006). Our proposal is similar to the pattern that models lists directly as chains of individuals but, for the

sake of clarity and standardization, we opted for using the properties names in the pattern that models lists as data structures.

Let $\mathcal{L}$ be a set of items $\{e_1, e_2, \ldots, e_n\}$ and $<$ a strict partial order, i.e. a binary relation that is irreflexive, transitive and asymmetric, defined for each pair $\langle e_i, e_j \rangle \in \mathcal{L} \times \mathcal{L}$, where $\times$ is the cartesian product. Based on the previous definitions, the basic concepts $List \sqsubseteq \top$ and $Item \sqsubseteq \top$ are defined, as well as the following properties:

$$hasNext \sqsubseteq isFollowedBy$$

*hasNext* is defined as a functional, asymmetric and irreflexive property, establishing the order of items in the list $\mathcal{L}$ according to the $<$ order. Due to the fact that it has been defined as a functional property just one item could follow another item. The inverse property is also defined as functional, forcing an item to be directly preceded by a unique item. The full definition of the $<$ order is achieved by introducing the transitive property *isFollowedBy* as a superproperty of *hasNext*. Since this means that *hasNext* implies *isFollowedBy*, any sequence of entities linked by *hasNext* will be inferred to be a chain linked by *isFollowedBy*. *hasNext* is used to express that an item *B* immediately follows another item *A*. There is no other item between them. So item *A* has *B* as the next item in the list (*A hasNext B*) or, in other words, item *A* is followed by item *B* (*A isFollowedBy B*). If another item *C* appears after item *B*, item *A* is also followed by item *C* (*A isFollowedBy C*), but item *A* has not *C* as the next item on the list (*not A hasNext C*). Furthermore, the property *hasItem* establishes the membership of an item in the list. *hasPrevious*, *isPrecededBy* and *isPartOf* are defined as the inverse properties of *hasNext*, *isFollowedBy* and *hasItem*, respectively.

$$hasPrevious \sqsubseteq isPrecededBy$$

The concepts *First* and *Last* identifies the starting and ending items of the list $\mathcal{L}$. Due to OWA in OWL, reasoners cannot automatically infer the individuals that belong to these concepts. Therefore, it is necessary to annotate these individuals when the text is represented.

$$First \equiv \neg(\exists\, hasNext^{-}\, \top)$$
$$Last \equiv \neg(\exists\, hasNext\, \top)$$

In OWL the same individual could be referred to in many different ways (i.e. with different URI references). Due to this, it is necessary to state that all the elements in the text are different individuals. For practical reasons, a functional property *hasID* is used to identify all of the individuals in the model with an unique code. In this way, the addition of new items to the ontology is easier, without the need of asserting that all of them are different from the existing individuals.

$$\top \sqsubseteq \forall\, hasID\, Datatype\#long$$

### 3.2 Basic concepts

Using the *list* concept defined above is possible to identify many of the entities that can be found in text documents, such as sentences or terms. Here, we present all the relevant entities in the model.

A sentence can be defined as a list of terms. Each term should be represented as a unique individual in the ontology regardless of its lexeme. The property *hasLexeme* is used to link a term with its lexeme.

$$\top \sqsubseteq \forall\, hasLexeme\, Datatype\#string$$
$$Term \sqsubseteq \leq 1\, hasLexeme$$
$$Sentence \equiv \exists\, hasElement\, Term$$

The proposed model defines forty commonly used Part of speechs (POS) (Santorini 1990). Actually, the set of POS used in the model are those defined in the Mark Watson's *Fast Tagger*,[5] which is used in our analyzer to annotate the terms according to their POS. The *Fast Tagger* library is based in turn on the Eric Brill's *Simple Rule-based Part of Speech Tagger* (Brill 1992).

The terms in a sentence can also be classified according to their gender and number. The *Feminine*, *Masculine* and *Neutral* concepts are defined as disjoint with each other. A term can therefore belongs to only one of those concepts. The *Plural* and *Singular* concepts are also defined as disjoint.

For efficiency reasons only some relations among individuals have to be established to represent the text. The rest of the relations can be inferred when necessary by adding a set of Description Logic axioms to the model without the need of processing the document again.

## 4 Methodology to build the discriminatory text analyzer

In this section, the methodology to build the basic text analyzer is proposed. The basic functional architecture of the system is depicted in Fig. 1. The text document is processed by the ontology populator module which first extracts all the tokens with the help of the *Fast Tagger* tokenizer function. Each of the sentences in the text is passed to the *Fast Tagger* tagger function which returns the most probable POS tag for each of the words in it. An individual is then created in the ontology for each token which is classified according to its POS or symbol category. A set of dictionaries are also used to classify the terms under the corresponding class in the ontology. There is a dictionary that assigns all sex specific occupations terms to the class *SexSpecificOccupation*, for instance. Then, Algorithm 1 is executed to establish all the necessary relations among entities in the text.

[5] https://github.com/mark-watson/fasttag_v2.

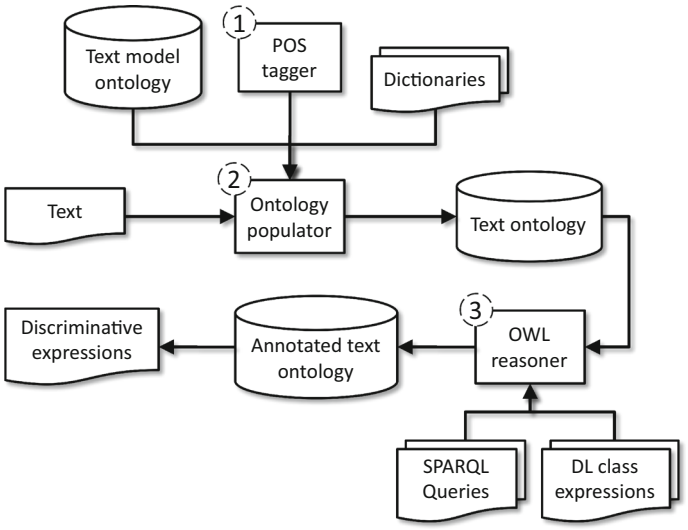


**Fig. 1** Functional architecture

**Algorithm 1** Load a document into the model

**Require:** $\mathcal{O}$ is the set of ontology axioms in DL and $\mathcal{D}$ is the document to be loaded (as a list of sentences)

**function** Tokenizer($\mathcal{O}, \mathcal{D}$)
  **for all** $s_i \in \mathcal{D}$ **do**
    $\mathcal{O} \leftarrow \mathcal{O} \cup \{s_i : Sentence\}$
    $last_s \leftarrow s_i$
    **for all** $t_j \in s_i$ **do**
      $\mathcal{O} \leftarrow \mathcal{O} \cup \{(t_j, nextID()) : hasID\}$
      $last_t \leftarrow t_j$
      $\mathcal{O} \leftarrow \mathcal{O} \cup \{(t_j, \phi(t_j)) : hasLexeme\}$
      **if** $j = 1$ **then**
        $\mathcal{O} \leftarrow \mathcal{O} \cup \{t_j : First\}$
        $\mathcal{O} \leftarrow \mathcal{O} \cup \{(t_j, s_i) : isPartOf\}$
      **else**
        $\mathcal{O} \leftarrow \mathcal{O} \cup \{(t_j, t_{j-1}) : hasPrevious\}$
    $\mathcal{O} \leftarrow \mathcal{O} \cup \{last_t : Last\}$
    **if** $i = 1$ **then**
      $\mathcal{O} \leftarrow \mathcal{O} \cup \{s_i : First\}$
      $\mathcal{O} \leftarrow \mathcal{O} \cup \{(s_i, d) : isPartOf\}$
    **else**
      $\mathcal{O} \leftarrow \mathcal{O} \cup \{(s_i, s_{i-1}) : hasPrevious\}$
  $\mathcal{O} \leftarrow \mathcal{O} \cup \{last_s : Last\}$
  **return** $\mathcal{O}$

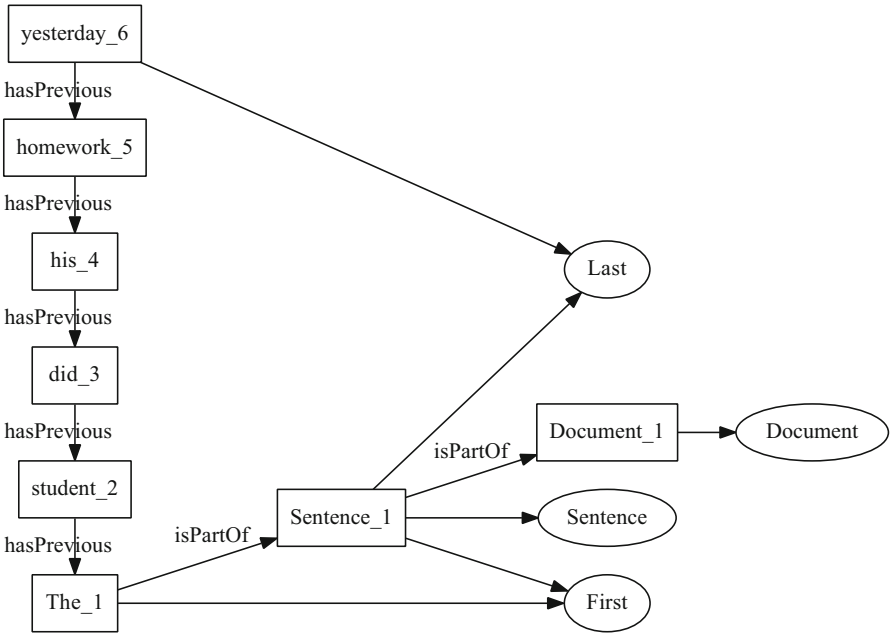


**Fig. 2** Assertions made by the Algorithm 1 for the sentence *"The student did his homework yesterday"*

Two kinds of DL axioms have been used in Algorithm 1. $\{c_i : C\}$ states that the individual $c_i$ belongs to a given concept $C$ whereas the axiom $\{(c_i, c_j) : P\}$ states that individuals $c_i$ and $c_j$ are related through the property $P$. So, the Line 3 of Algorithm 1 just creates the individual $S_i$ for representing the $i$-th sentence of the text. For each term in the sentence, a new individual is created in the ontology. As discussed in Sect. 3.2, each term has an associated unique identifier. For this purpose, the function *nextID()* from line 6 is used with the aim that reasoners identify all terms as different individuals. To do so, the property *hasID()* has been defined to be a functional relation.

The $\phi(w)$ function, on line 8 of Algorithm 1, returns the lexeme for the term $w$. It is not necessary to create the terms as instances of the concept *Term* because it is the domain of the property *hasLexeme*. Therefore, the reasoner will automatically identify them as individuals of the concept *Term*.

In order to relate all terms to their sentences, the property *isPartOf* is used. In our methodology, it is not necessary to relate all tokens to the sentence they belong to because the tokens of a sentence are part of the same sentence that the previous previous terms are also part of. Therefore, just the first term of the sentence needs to be associated to the sentence of which it is a member.

The statement on line 13 establishes the order relation among elements of the sentence, relating two terms by means of the property *hasPrevious*, considering that the only term without previous element is the first term in the sentence.

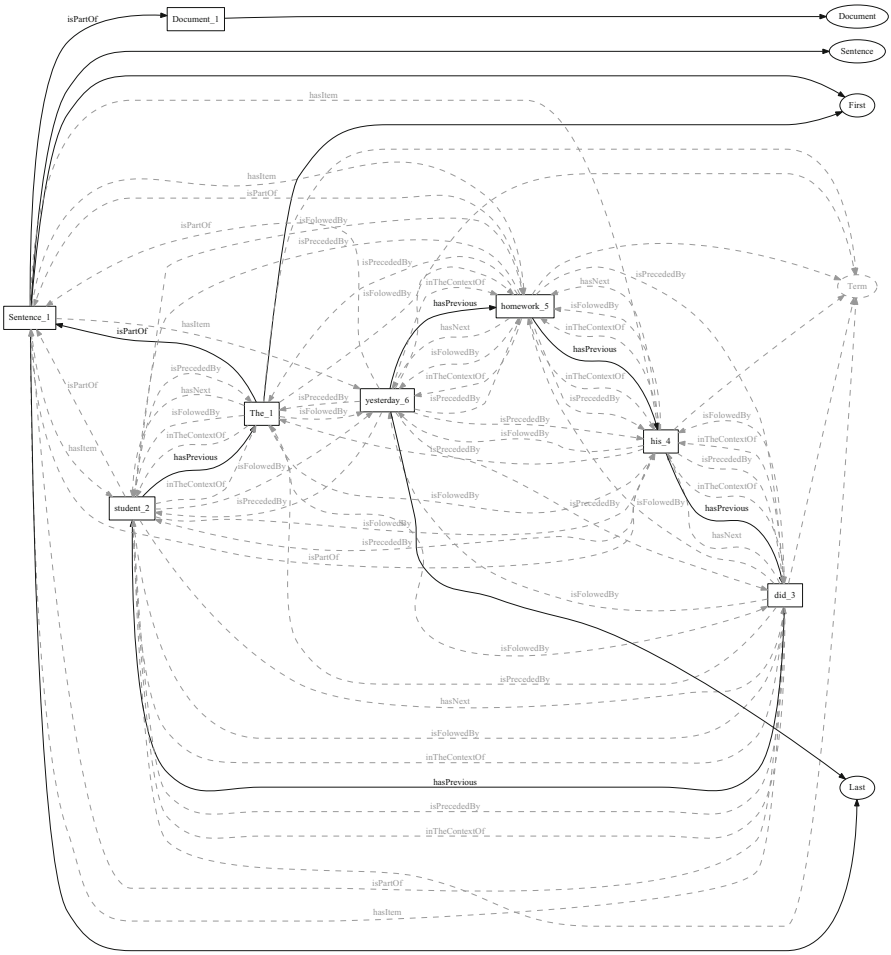


**Fig. 3** Inferred knowledge for the sentence *"The student did his homework yesterday"*

Therefore, we can see that the proposed methodology requires minimum relations to represent the text. If needed, the rest of the relations can be inferred through a reasoning process to analyze the text. To illustrate this fact, a simple illustrative example is provided in Figs. 2 and 3.

Figure 2 shows the assertions made by the Algorithm 1 to process the sentence *"The student did his homework yesterday"*. Individuals appear as rectangles in the diagram whereas concepts are drawn as circles. For the sake of clarity, the data types properties *hasLexeme* and *hasID* have been hidden as well as the label for the RDF *type* relation. The arrows connecting "Sentence_1" with classes "First" and "Last" indicate that it is the first and the last sentence in the text, respectively.

From the assertions indicated in Fig. 2 the set of relations that are shown in Fig. 3 are inferred. Those inferred relations can be used in rules in order to identify new patterns without the need of modifying the algorithm for loading the document

nor processing the document again. In fact, those inferred relations are computed just in the case they are needed. Our proposal for the analysis of texts exploit the full reasoning capabilities of ontologies, inferring knowledge, which is one of the main advantages of ontologies.

All the reasoning processes are performed by the OWL reasoner. The *mini OWL rules inference engine* of JENA API[6] is enough for computing all the inferences. We choose this API because it allows us to easily work with SPARQL queries, which are used to identify discriminative expressions in text. DL class expressions can also be used to find that kinds of expressions. Both SPARQL queries and DL class expressions are used to classify discriminative terms under the corresponding concept in the ontology. A masculine pronoun that is used to refers to both men a women may be classified as an individual of the concept *Invisibility*, for instance.

## 5 An ontology-based analyzer for the detection of discriminatory language

In this section, we describe the ontology-based analyzer that we have developed by using the proposed ontology-based document text model in order to detect some discriminatory expressions. To do so, first we propose the representation of the incidences in the model that includes general common rules for detecting incidences and the specific rules in the fields of gender and disability. Then, the discriminatory language text editor we have developed by using the ontology-based analyzer for the detection of discriminatory language is also presented.

### 5.1 Representing incidences in the model

An incidence in the model represents a potential discriminatory expression. For the detection of such expressions we have used as base the Inclusive Language Guideline (University of Newcastle 2006). We have extracted a set of rules from this document, detailed in Appendix 2. More rules can be added to the ontology to detect other discriminatory expressions without the need of modifying the analyzer code. This is the main advantage of our proposal, which offers a flexible and reusable model.

A set of concepts have been defined in the model for the representation of those incidences. All those concepts represent a kind of discriminatory expression and the reasoner is responsible for making the automatic classification of the terms in any of those concepts. Because all of them have been defined as subconcept of the *Incidence* concept, the analyzer just need to get all the individuals belonging to this concept to find out all the potential discriminatory expressions.

In some cases, a concept representing a kind of incidence is just a set of forbidden terms that must not be used. The *hasLexeme* property can be used in such cases to identify the set of forbidden words. Although this is a perfectly valid alternative, it is important to note that the performance of OWL reasoners is very

[6] https://jena.apache.org.

poor when dealing with strings. A text document with hundred of terms requires few minutes to be processed in this way. A more convenient alternative consists on creating a set of concepts in the model to represent the terms having a relevant lexeme and annotate them when the document is processed. This alternative is far faster than the former and the way the analyzer in this proposal has been implemented. The *Man* concept is defined in the model because the term *man* is relevant in this context, for instance. It is a relevant term because it is used as a key word in some of the rules defined in the following section.

In most cases, a concept representing an incidence is defined as a pattern of terms, which may include references to the terms in another sentences. The patterns are defined using the list structure introduced in Sect. 3.1.

#### *5.1.1 Common rules for detecting discriminatory language*

Although there are some specific rules for detecting potential discriminatory expressions, there exist some common rules that can be used for the recognition of the major forms of discriminatory language. These rules usually try to avoid the use of expressions that unnecessarily remark the differences among groups of people.

It is usually unnecessary to mention a person's sex, race, ethnic background, religion or disability. These characteristics are often used in discriminatory language when describing members of minority groups. To avoid such kinds of expressions we have created the *Extravisibility* concept which comprises the nouns in the document being preceded by any term related to those characteristics.

$$\begin{aligned} &Extravisibility \sqsubseteq Exclusive \\ &Extravisibility \equiv (DisabledPeople \sqcup RacePeople \sqcup ReligionPeople \sqcup \\ &\qquad Sex) \sqcap \exists\, hasNext\, Noun \end{aligned}$$

The terms in the document are annotated with their corresponding POS tag using the *Fast Tagger* library. A dictionary based approach is used to annotate the terms according to their meaning. *Autistic*, *Blind* or *Deaf* terms are annotated as *DisabledPeople*, for instance.

#### *5.1.2 Rules for gender neutral language*

The gender is one the main sources of discriminatory language, so many specific rules have been proposed to avoid such kind of language in this context. Actually, women are often invisible in language due to the use of the masculine pronouns to refer to both men and women. In order to detect this issue the proposed model is extended with a new concept called *NeutralMasculinePronoun* which represents the masculine pronouns that are used correctly in the text. In the context of gender language, the singular masculine pronoun is used correctly when referring to a particular entity or when used in conjunction with the singular feminine pronoun. The *NeutralMasculinePronoun* concept represents those pronouns. Such kind of individuals in the ontology can be automatically inferred by defining the following equivalences.

$$NeutralMasculinePronoun \equiv Pronoun \sqcap Masculine \sqcap \exists hasNext\,(\exists hasNext\,(Pronoun \sqcap Feminine)) \quad (1)$$

$$NeutralMasculinePronoun \equiv Pronoun \sqcap Masculine \sqcap \exists isPrecededBy\,ProperNoun \quad (2)$$

The Axiom 1 identifies the individuals belonging to the concept *NeutralMasculinePronoun* as those masculine pronouns that are followed by a term, which in turn is followed by a feminine pronoun. This is the case of the masculine pronoun *he* in the sentence *"If a student studies hard, he or she will succeed"*. The term *he* is followed by the term *or*, which in turn is followed by a feminine pronoun.

The Axiom 2 is used to determine when the masculine pronouns are used to refer to an entity previously introduced in the sentence. The masculine pronoun in the sentence *"If John studies hard, he will succeed"* is matched by this rule because of the proper noun previously introduced in the sentence.

All those masculine pronouns not belonging to the concept *NeutralMasculinePronoun* are considered as incidences. Due to the OWA in OWL, defining a complementary concept to find these issues is not enough. It is necessary to use a more convenient query language such as SPARQL. The query shown below annotates all the masculine pronouns not being matched by Axioms 1 or 2 as individuals of the concept *Invisibility*.

```
CONSTRUCT { ?term a Invisibility . }
WHERE {
  ?term a Masculine .
  ?term a Pronoun .


  FILTER NOT EXISTS { ?term a NeutralMasculinePronoun . }
}
```

The above query finds out masculine pronouns not being identified as neutral, such as the masculine pronoun in the sentence *"If a student studies hard, he will succeed"*. In this case, the pronoun is not preceded by a proper noun neither has a feminine pronoun close to it. Following this process the analyzer is able to detect some of the issues described in Rules 3.1.1 and 3.2.4 of Appendix 2.

Co-reference resolution may play an important role in these kind of rules, where masculine pronouns are used as generic pronouns. Being able to determine the gender of the entity the pronoun is referring to may improve the performance of the text analyzer. However, as the most common cause of the occurrence of these rules is the usage of a masculine pronoun to refer to a neutral entity introduced previously

in the sentence, the Axiom 2 may be enough for detecting most of the incidences without adding more complexity to the text analyzer.

The Rule 3.2.2 tries to prevent the use of the term *man* as a verb. To detect those incidences the concept *ManAsVerb* is defined as the following equivalence.

$$Man \equiv hasLexeme\ value\ "Man" \tag{3}$$

$$Manning \equiv hasLexeme\ value\ "Manning" \tag{4}$$

$$\begin{aligned} The &\equiv hasLexeme\ value\ "The" \\ ManAsVerb &\sqsubseteq Exclusive \\ ManAsVerb &\equiv (Man \sqcup Manning) \sqcap \exists\, hasNext\ The \end{aligned} \tag{5}$$

All the terms with lexemes *man* or *manning* followed by the term *the* will be classified by the reasoner as individuals of the concept *ManAsVerb*. It is noteworthy that the inclusion of the Axioms 3, 4 and 5 in the ontology could adversely affect in the performance of the reasoner, as explained previously. *Man*, *manning* and *the* are considered relevant terms and are annotated while processing the text document, improving the efficiency of the reasoning.

A similar scheme is followed to implement the Rules 3.1.2, 3.2.1, 3.2.2 and 3.2.6.

$$\begin{aligned} Sex &\equiv hasLexeme\ value\ "Woman" \sqcup hasLexeme\ value\ "Man" \\ &\quad \sqcup hasLexeme\ value\ "Female" \sqcup \ldots \\ Inferiority &\equiv Sex \sqcap \exists\, hasNext\ Profession \\ For &\equiv hasLexeme\ value\ "For" \\ In &\equiv hasLexeme\ value\ "In" \\ Of &\equiv hasLexeme\ value\ "Of" \\ ManAlternative &\equiv Man \sqcap \exists\, hasNext\ (For \sqcup In \sqcup Of) \\ Men &\equiv hasLexeme\ value\ "Men" \\ Women &\equiv hasLexeme\ value\ "Women" \\ Sir &\equiv hasLexeme\ value\ "Sir" \\ Madam &\equiv hasLexeme\ value\ "Madam" \\ &\ldots \\ MenWomenOrder &\equiv (Men \sqcap \exists\, hasNext\ (\exists\, hasNext\ Women)) \sqcup (Sir \sqcap \\ &\quad \exists\, hasNext\ (\exists\, hasNext\ Madam)) \sqcup \ldots \end{aligned}$$

*Inferiority*, *ManAlternative* and *MenWomenOrder* concepts are created as subconcept of *Exclusive*. The *MenWomenOrder* concept is another example about how the list structure described in Sect. 3.1 is used to represent the textual data. It matches all the terms referring to men that are followed by a term which, in turn, is

followed by a term referring to women. This rule will be used to suggest the user to vary the word order of those terms regularly.

There are many rules that make use of the list structure intensively. For example, to avoid sexist descriptions it is important to take care of gender generalizations and descriptions, as suggested by Rule 3.2.12 in Appendix 2. As a specific case of those exclusive expressions, the analyzer will prevent the use of expression having different adjectives to describe men and women. The concept *SexistDescription* match all those adjectives followed by the term *women* and preceded by the terms *and* and *men*. The term *men* should also be preceded by an adjective to be recognized as an individual belonging to the concept *SexistDescription*.

$$\begin{aligned} And &\equiv hasLexeme\ value\ ''And'' \\ SexistDescription &\equiv Adjective \sqcap \exists hasNext\ Women \\ &\quad \sqcap \exists hasPrevious\,(And \sqcap \exists hasPrevious\,(Men \sqcap \\ &\quad \exists hasPrevious\ Adjective)) \end{aligned}$$

The analyzer uses a similar approach to implement the Rule 3.2.8. In this case the analyzer ensures that no feminine, proper noun is used when the terms *Mr* or *Dr* is used. The concept *InappropriateTitles* matches all those feminine, proper nouns.

$$\begin{aligned} InappropriateTitles \equiv\ & (Dr \sqcup Mr) \sqcap \exists hasNext\,(\exists hasNext\,(Feminine \\ & \sqcap ProperNoun)) \\ & \sqcup \exists hasNext\,(\exists hasNext\,(\exists hasNext\,(Feminine \sqcap \\ & ProperNoun))) \\ & \sqcup \exists hasPrevious\,(\exists hasPrevious\,(Feminine \sqcap \\ & ProperNoun)) \\ & \sqcup \exists hasPrevious\,(\exists hasPrevious\,(\exists hasPrevious \\ & (Feminine \sqcap ProperNoun))) \end{aligned}$$

To avoid patronizing expressions the words *man/woman*, *girl/boy*, *gentleman/lady* must be used in a parallel manner, as pointed out by Rule 3.2.10 in B. To implement this rule the analyzer first tries to detect the terms that are used correctly. Any of those terms is used correctly if its equivalent for the opposite gender is also used in the context. The transitive properties *isPrecededBy* and *isFollowedBy* can be used in this case. The analyzer also tries to find the equivalent of the term in the previous or the next sentence. For simplicity, just the first term in the sentence is associated to that sentence. It is used by the reasoner to get the previous and the next sentence. As this pattern is slightly more complex to be expressed in DL, we have chosen SPARQL to implement this rule. The following query has to be executed to get the terms *girl* that are used correctly.

```
CONSTRUCT { ?term a notPatronising . }
WHERE {
  { ?term a Girl ;
    isPrecededBy ?term2 .
    ?term2 a Boy .
  } UNION { ?term a Girl ;
    isPrecededBy ?term2 .
    ?term2 a Boy .
  } UNION { ?term a Girl ;
    isPrecededBy ?first .
    ?first a First .
    ?first isPartOf ?sent .
    ?senthasPrevious ?prsent .
    ?prsent hasItem ?prfirst .
    ?prfirst isFollowedBy ?term2 .
    ?term2 a Boy .
  } UNION { ?term a Girl ;

    isPrecededBy ?first .
    ?first a First .
    ?first isPartOf ?sent .
    ?senthasNext ?nxsent .
    ?nxsent hasItem ?nxfirst .
    ?nxfirst isFollowedBy ?term2 .
    ?term2 a Boy .
  }
}
```

Equivalent queries are used with the rest of the terms (*man*/*woman*, *gentleman*/*lady*...). The terms not being matched by this rule are considered to be exclusive terms because they are not used in a parallel manner. Again, due to OWA in OWL, it is necessary to use the SPARQL language to find them, as shown bellow.

```
CONSTRUCT { ?term a Patronising . }
WHERE {
  ?term a Girl .
  FILTER NOT EXISTS { ?term a notPatronising . }
}
```

There are also some rules in Appendix 2 based on a list of forbidden terms, such as Rules 3.2.1, 3.2.7, 4.1.1 and 4.1.2. For efficiency reasons, all the terms in those lists are considered relevant terms and annotated while processing the document.

### *5.1.3 Language and people with disabilities*

To show how easily the proposed model can be extended, we present in this section some of the rules that should be included to detect discriminatory language in relation to the portrayal of people with disabilities.

The DL axioms describing these rules could be defined in another OWL file and loaded by the analyzer if needed. In such case, the text does not need to be processed again. The reasoner will classify the existing individuals according to the new class descriptions. This way, the analyzer functionality can be programmed in a modular way.

Apart from the rules based on a list of forbidden terms, such as Rules 4.1.1 and 4.1.2, the Rule 4.1.3 suggests that the terms *victim* or *sufferer* must never be used to refer to a person who has or has had an illness, disease or disability. The *Stereotyping* concept defined below represents those terms.

$$\begin{aligned} Sufferer &\equiv hasLexeme\ value\ ''Sufferer'' \\ Victim &\equiv hasLexeme\ value\ ''Victim'' \\ Illness &\equiv hasLexeme\ value\ ''Alzheimer'' \\ &\sqcup\ hasLexeme\ value\ ''Anorexia''\ \sqcup\ \ldots \\ Stereotyping &\equiv\ (Sufferer\ \sqcup\ Victim)\ \sqcap\ (\exists\, isFollowedBy\, Illness \\ &\sqcup \exists\, isPrecededBy\, Illness) \end{aligned}$$

## 5.2 Comparative results

To evaluate the performance of our proposal we have created a small data set containing six hundred and twenty two sentences. Discriminatory sentences have been extracted from various non-discriminatory language guides and other documents found on Internet.[7] For each of the sentences in this basic data set some variations have been added, where some of their terms have been changed or rearranged. In some cases we tried to preserve the meaning of the sentences whereas in other cases we tried to form a completely different sentence with roughly the same words. When possible, the data set also includes the corresponding non-discriminatory expressions. We decided to not include sentences containing forbidden words, such as *cripple*, because those type of incidences can be easily identified by text analyzers. Instead, the data set contains sentences with complex discriminatory expressions, such as those shown in Sect. 5.1.

Table 1 shows the number of sentences in the data set according to the reason they have been annotated as discriminatory. The derived non-discriminatory expressions are also shown on the table. All of them have been added to a global data set. Please note that there are some sentences that may belong to more than one category, but only one copy of them has been added, so there are no duplicated

[7] A Weka-ready version of the data set is available at https://sourceforge.net/p/disclangeditor/.

sentences in the global data set. The total number of sentences passed to classifiers is shown at the end of the table.

The entire data set has been used as the test set for evaluating our proposal. However, because the rest of the classifiers are based on supervised learning approaches, a 10-fold cross-validation has been used. Sentences have been converted to vectors using the *StringToWordVector* Weka filter. To evaluate the impact of the size of the n-grams in the performance of classifiers, unigrams (one word), bigrams (two words) and trigrams (three words) versions of the classifiers were included in the analysis. Stop words have been removed from the data set for all classifiers. Table 2 shows the performance of our proposal compared to some classifiers in the Weka data mining tool. The results have been grouped by the size of the n-gram and then sorted by the weighted average f-measure value.

The results show the performance of the proposed model. It achieves the highest overall classification performance, followed by the *Simple logistic* and *Binary SMO* classifiers. It also gets by far the highest performance score for detecting discriminatory sentences although it is a bit overcome by the *Simple logistic* classifier when classifying sentences as non-discriminatory due to a relatively low recall value. Clearly, the problem with the *Simple logistic* classifier is the low recall when classifying discriminatory sentences. There are four classifiers that clearly seem not appropriate for detecting discriminatory language: *1R*, *Bayes Network*, *RBF Network* and *Conjunctive rules*. All of them produce correct predictions when they found a sentence to be discriminatory, but the number of those predictions is extremely low.

With respect to the size of the n-grams, results indicate that the unigram alternative is better for addressing this problem. The higher the size of the n-gram the worse is the performance of all the classifiers. This is probably due to the complexity of the patterns that make sentences discriminatory. A term may affect the classification of the sentence depending on its relative position with respect to another term, for instance. The *Decision Table* is the classifier that is less affected by the size of the n-gram, but its overall performance is also degraded as the size is increased.

The main drawback of our proposal is, however, the time needed to process the text. Reasoners require a lot of time to compute the inferences. It may takes up to thirty seconds to compute all the inferences for a text, although subsequent SPARQL queries take a few milliseconds to complete. This is because the inferences for the *TBOX*[8] are shared among sentences. For this reason, it is far more efficient to load a paragraph or even the entire text document than analyzing each sentence individually. The editor presented in this paper uses a multithreaded approach to process the text in the background while the user is typing, in the same way as professional office suits check the orthography and grammar of documents, for instance. The delay is only noticeable when the user loads a long document.

On the other hand, most of the Weka classifiers took no more than five or six seconds to evaluate the entire data set. There are, however, some exceptions such as the *Simple Logistic* classifier—the second best classifier—which took eighty

[8] Concepts and properties of the ontology.

seconds to build the model and fifteen minutes to evaluate the entire data set. This time is still far below the hour that took our proposal for the same task. However, it is important to note that we have considered all sentences to be independent documents for testing our classifier, as the other classifiers do, so the full reasoning process is repeated for all the sentences, even sharing the same *TBOX*. We cannot analyze the entire data set at a time because our proposal uses information on surrounding sentences to improve its performance.

### 5.3 Discriminatory language editor

A text editor in Java have been released[9] under the GPL open source license in order to facilitate the use of the proposed methodology. The text editor is able to detect discriminatory expressions while the user is typing. When the internal analyzer detects a potential discriminatory expression the user is advised by underscoring the related words in the text, as shown in Fig. 4. Furthermore, a descriptive message about the issue is also shown to the user when the cursor is placed over the potential discriminatory expression.

The editor can also be used as a command line application. In this case, the text document to be analyzed should be passed to the application as an argument. As result, the application generates an ontology, specified in our proposed model, representing the document that has been processed.

## 6 Conclusions and future works

A key issue in the last decades is the discriminatory language like gender, race, ethnicity, etc. Text analyzers that detect such kind of language are needed to promote equity in society in the use of language, which is flexible and, usually, contains hidden features or relations. However, there is no flexible tool for analyzing texts that can be easily adapted to detect different kinds of discrimination.

The main objective of our proposal is to show how ontologies can be used to build a rule-based discriminative text analyzer. Patterns that are based on the presence of multiple terms regardless of their order are better recognized by text analyzers based on the n-gram approach. However, regular expressions perform better when the order of the terms matters. The text analyzer presented in this work is based on the regular expressions approach but it is using DL class expressions and SPARQL queries to express them. The main advantage of using ontologies is the flexibility they provide for building those expressions because most of the relations among the entities in the text can be inferred by a reasoner without the need of building a specific application to make them explicit.

Ontologies also provide a great flexibility for representing the text and they have been designed to be easily combined, incorporating new knowledge to the system. We are working on the incorporation of a word sense disambiguation module that allows us include information about the meaning of terms in the rules.

[9] https://sourceforge.net/p/disclangeditor/.

**Table 1** Data set structure

| Category | Issue | Discriminatory sentences | | |
|---|---|---|---|---|
| | | Yes | No | Total |
| General | Extravisibility | 44 | 69 | 113 |
| Gender | Invisibility | 57 | 84 | 141 |
| | Inferiority | 33 | 47 | 80 |
| | Man alternatives | 25 | 30 | 55 |
| | Man as a verb | 26 | 32 | 58 |
| | Sex-specific occupations | 25 | 27 | 52 |
| | Inappropriate titles | 27 | 16 | 43 |
| | Use of 'Ms' | 17 | 19 | 36 |
| | Patronising expressions | 33 | 36 | 69 |
| | Sexist descriptions | 18 | 2 | 20 |
| Disability | Derogatory labelling | 26 | 26 | 52 |
| | Depersonalising | 24 | 25 | 49 |
| | Stereotyping | 16 | 19 | 35 |
| | | 287 | 335 | 622 |

To show the usefulness of our proposal we have developed a basic text analyzer software that is able to detect some discriminatory expressions. We have shown how the kernel of the model can be extended with domain specific concepts and how the proposed methodology should be applied to documents in order to find out relevant patterns. We have extended the set of rules of the text analyzer with several new rules in order to detect some discriminatory expressions in terms of gender and people with disabilities. A data set has been developed and it has been used to evaluate the performance of various classifiers for detecting discriminatory sentences. The results show that the overall performance of our proposal is better than the performance of the other classifiers. In fact, we consider that the performance of our proposal is even higher than the obtained in the analysis because we have considered all sentences to be independent documents for testing all the classifiers. However, in a real scenario, our proposal is able to use the information on surrounding sentences for improving its performance.

One of the drawbacks of using ontologies as the base of text analyzers is that, depending on the number of terms and sentences in the text, the reasoning process for the detection of relevant patterns may be slightly slow. We have pointed out solutions that we have adopted in order to improve the efficiency of the analyzer.

The Discriminative Language Editor presented in this paper, is our first user friendly application, but it is actually part of a bigger project that is focused on using ontologies to deal with texts.[10] A slightly modified version of the proposed model has been successfully used in the field of opinion mining to find the sentiment polarities of documents (Salguero and Espinilla 2016). Instead of manually defining a set of rules to process the text documents, a DL Class Expression Learner (CEL) is

[10] http://sinbad2.ujaen.es/text-mining-dist.

**Table 2** Performance of classifiers

| | Discriminatory:yes | | | Discriminatory:no | | | Weighted Avg. | | |
|---|---|---|---|---|---|---|---|---|---|
| | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ |
| 1-gram | | | | | | | | | |
| Our proposal | 0.68 | 0.83 | 0.75 | 0.82 | 0.67 | 0.74 | 0.76 | 0.74 | 0.74 |
| Simple logistic | 0.76 | 0.60 | 0.67 | 0.71 | 0.83 | 0.77 | 0.73 | 0.73 | 0.72 |
| Binary SMO | 0.67 | 0.66 | 0.67 | 0.71 | 0.72 | 0.72 | 0.70 | 0.70 | 0.70 |
| SPegasos | 0.65 | 0.62 | 0.64 | 0.69 | 0.71 | 0.70 | 0.67 | 0.67 | 0.67 |
| BL regression | 0.60 | 0.67 | 0.63 | 0.68 | 0.63 | 0.65 | 0.65 | 0.64 | 0.64 |
| Decision Table | 0.79 | 0.36 | 0.49 | 0.63 | 0.92 | 0.74 | 0.70 | 0.66 | 0.63 |
| J48 pruned tree | 0.77 | 0.34 | 0.47 | 0.62 | 0.91 | 0.73 | 0.69 | 0.65 | 0.61 |
| Voted perceptron | 0.56 | 0.56 | 0.56 | 0.62 | 0.62 | 0.62 | 0.60 | 0.60 | 0.60 |
| Naive Bayes | 0.55 | 0.48 | 0.52 | 0.60 | 0.66 | 0.63 | 0.58 | 0.58 | 0.58 |
| 1R | 1.00 | 0.08 | 0.15 | 0.56 | 1.00 | 0.72 | 0.76 | 0.58 | 0.46 |
| Bayes Network | 1.00 | 0.08 | 0.15 | 0.56 | 1.00 | 0.72 | 0.76 | 0.58 | 0.46 |
| RBF network | 1.00 | 0.00 | 0.00 | 0.54 | 1.00 | 0.70 | 0.76 | 0.54 | 0.38 |
| Conjunctiverules | 1.00 | 0.01 | 0.01 | 0.54 | 1.00 | 0.70 | 0.75 | 0.54 | 0.38 |
| 2-gram | | | | | | | | | |
| Simple logistic | 0.79 | 0.25 | 0.39 | 0.60 | 0.94 | 0.73 | 0.69 | 0.63 | 0.57 |
| Decision Table | 0.84 | 0.22 | 0.35 | 0.59 | 0.96 | 0.73 | 0.70 | 0.62 | 0.56 |
| Binary SMO | 0.50 | 0.46 | 0.48 | 0.57 | 0.60 | 0.58 | 0.53 | 0.54 | 0.53 |
| SPegasos | 0.47 | 0.44 | 0.46 | 0.55 | 0.58 | 0.57 | 0.51 | 0.51 | 0.51 |
| BL regression | 0.44 | 0.45 | 0.44 | 0.52 | 0.52 | 0.52 | 0.49 | 0.49 | 0.49 |
| Voted perceptron | 0.44 | 0.40 | 0.42 | 0.53 | 0.57 | 0.55 | 0.49 | 0.49 | 0.49 |
| Naive Bayes | 0.38 | 0.32 | 0.35 | 0.49 | 0.56 | 0.52 | 0.44 | 0.45 | 0.44 |
| Bayes Network | 1.00 | 0.05 | 0.01 | 0.55 | 1.00 | 0.71 | 0.76 | 0.56 | 0.43 |
| 1R | 1.00 | 0.05 | 0.01 | 0.56 | 1.00 | 0.71 | 0.76 | 0.56 | 0.43 |
| Conjunctiverules | 0.00 | 0.00 | 0.00 | 0.54 | 1.00 | 0.70 | 0.29 | 0.54 | 0.38 |
| RBF network | 1.00 | 0.01 | 0.01 | 0.54 | 1.00 | 0.70 | 0.75 | 0.54 | 0.38 |
| J48 pruned tree | 0.24 | 0.01 | 0.03 | 0.53 | 0.96 | 0.69 | 0.40 | 0.52 | 0.38 |
| 3-gram | | | | | | | | | |
| Decision Table | 0.82 | 0.08 | 0.14 | 0.56 | 0.99 | 0.71 | 0.68 | 0.57 | 0.46 |
| Simple logistic | 0.66 | 0.09 | 0.15 | 0.56 | 0.96 | 0.70 | 0.60 | 0.56 | 0.45 |
| Voted perceptron | 0.37 | 0.24 | 0.29 | 0.50 | 0.65 | 0.57 | 0.44 | 0.46 | 0.44 |
| 1R | 1.00 | 0.04 | 0.07 | 0.55 | 1.00 | 0.71 | 0.76 | 0.56 | 0.41 |
| Binary SMO | 0.30 | 0.24 | 0.27 | 0.45 | 0.53 | 0.49 | 0.38 | 0.40 | 0.39 |
| Conjunctiverules | 0.00 | 0.00 | 0.00 | 0.54 | 1.00 | 0.70 | 0.29 | 0.54 | 0.38 |
| Bayes Network | 0.00 | 0.00 | 0.00 | 0.54 | 1.00 | 0.70 | 0.29 | 0.54 | 0.38 |
| J48 pruned tree | 0.00 | 0.00 | 0.00 | 0.54 | 1.00 | 0.70 | 0.29 | 0.54 | 0.38 |
| RBF network | 1.00 | 0.01 | 0.01 | 0.54 | 1.00 | 0.70 | 0.75 | 0.54 | 0.38 |
| SPegasos | 0.23 | 0.23 | 0.26 | 0.44 | 0.52 | 0.48 | 0.37 | 0.38 | 0.37 |
| Naive Bayes | 0.25 | 0.19 | 0.21 | 0.43 | 0.53 | 0.48 | 0.35 | 0.37 | 0.36 |

**Table 2** continued

| | Discriminatory:yes | | | Discriminatory:no | | | Weighted Avg. | | |
|---|---|---|---|---|---|---|---|---|---|
| | *P* | *R* | $F_1$ | *P* | *R* | $F_1$ | *P* | *R* | $F_1$ |
| BL regression | 0.30 | 0.33 | 0.31 | 0.37 | 0.33 | 0.35 | 0.34 | 0.33 | 0.33 |

*P* precision, *R* recall, $F_1$ F-measure

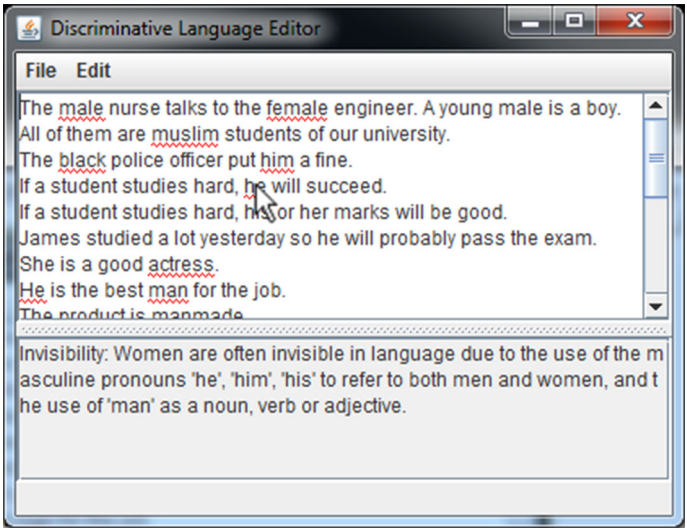


**Fig. 4** Discriminatory language editor

used to automatically infer class expressions that characterize documents expressing positive opinions.

Co-reference resolution may play an important role not only in some of the rules of the text analyzer but also in many other fields of text analysis. For this reason, we are also working in an extension of the model for co-reference resolution, which is also available in the web of the master project. Preliminary results show a small improvement over other popular co-reference detectors such as JavaRAP or the Standford CoreNLP suite. The Semantic Web Rule Language is used in this extension to make the n-grams of a given size explicit, so it can be used in rules or CEL algorithms to improve the performance of text classifiers.

**Acknowledgements** This contribution has been supported by the Andalusian Institute of Women, Junta de Andalucía, Spain (Grant No. UNIVER09/2009/23/00).

## Appendix 1: Relevant class descriptions for discriminative language detection

Extra-visibility ≡ DisabledPeople ⊔ RacePeople ⊔ ReligionPeople ⊔ Sex ⊓ ∃ hasNext Noun

InappropriateTitles ≡ Dr ⊔ Mr ⊓ ∃ hasNext (∃ hasNext (Feminine ⊓ ProperNoun)) ⊔ ∃ hasNext (∃ hasNext (∃ hasNext (Feminine ⊓ ProperNoun))) ⊔ ∃ hasPrevious (∃ hasPrevious (Feminine ⊓ ProperNoun)) ⊔ ∃ hasPrevious (∃ hasPrevious (∃ hasPrevious (Feminine ⊓ ProperNoun))) ⊔ Mrs ⊔ Ms ⊓ ∃ hasNext (∃ hasNext (Masculine ⊓ ProperNoun)) ⊔ ∃ hasNext (∃ hasNext (∃ hasNext (Masculine ⊓ ProperNoun))) ⊔ ∃ hasPrevious (∃ hasPrevious (Masculine ⊓ ProperNoun)) ⊔ ∃ hasPrevious (∃ hasPrevious (∃ hasPrevious (Masculine ⊓ ProperNoun)))

ManAsVerb ≡ Man ⊔ Manning ⊓ ∃ hasNext The

ManPrecededByForInOf ≡ Man ⊓ ∃ hasNext (For ⊔ In ⊔ Of)

ManPrecededByForInOf ⊑ ManAlternative

MenWomenOrder ≡ He ⊓ ∃ hasNext (∃ hasNext She) ⊔ Him ⊓ ∃ hasNext (∃ hasNext Her) ⊔ His ⊓ ∃ hasNext (∃ hasNext Hers) ⊔ Men ⊓ ∃ hasNext (∃ hasNext Women) ⊔ Sir ⊓ ∃ hasNext (∃ hasNext Madam)

NeutralMasculinePronoun ≡ Masculine ⊓ Pronoun ⊓ ∃ isPrecededBy ProperNoun

NeutralMasculinePronoun ≡ Masculine ⊓ Pronoun ⊓ ∃ hasNext (∃ hasNext (Feminine ⊓ Pronoun))

SexistDescription ≡ Adjective ⊓ ∃ hasNext Women ⊓ ∃ hasPrevious (And ⊓ ∃ hasPrevious (Men ⊓ ∃ hasPrevious Adjective))

Stereotyping ≡ Sufferer ⊔ Victim ⊓ ∃ isFollowedBy Illness ⊔ ∃ isPrecededBy Illness

## Appendix 2: Rules for detecting discriminative language

| Rule | Description | Examples | |
|---|---|---|---|
| 2.1. Extra-visibility | It is quite unnecessary to mention a person's sex, race, ethnic background, religion or disability | Male nurse; female engineer; muslim student; Black police officer | ✓ |
| 3.1.1. Invisibility | Women are often invisible in language due to the use of the masculine pronouns 'he', 'him', 'his' to refer to both men and women, and the use of 'man' as a noun, verb or adjective | Mankind; man made | ✓ |
| 3.1.2. Inferiority | Unnecessary mention of gender to suggest that in certain roles women are inferior to men. The use of 'feminine' suffixes such as 'ette', 'ess', 'ienne' and 'trix' are unnecessary | Female engineer; woman academic; actress | ✓ |
| 3.2.1. Use alternatives for 'man' | | Mankind; the best man for the job; the man in the street; man of letters, men of science; manpower; manmade | ✓ |
| 3.2.2. Avoid the use of 'man' as a verb | | We need someone to man the desk; manning the office; She will man the phones | ✓ |
| 3.2.4. Find alternatives to 'he' and 'his' | The pronouns 'he', 'his' and 'him' are frequently used as generic pronouns. As this use is both ambiguous and excludes women, try to find alternatives | The student may exercise his right to appeal | ✓ |
| 3.2.7. Use alternatives for sex-specific occupation terms | Avoid the impression that these positions are male-exclusive. Avoid using occupational titles containing the 'feminine' suffixes -ess, -ette, -trix, -ienne. | Chairman; headmaster; headmistress; policeman; businessman; layman; groundsman; actress; executrix; authoress; comedienne | ✓ |
| 3.2.8. Use appropriate titles and other modes of address | The inappropriate use of names, titles, salutations and endearments create the impression that women merit less respect or less serious consideration that men do. Ensure that people's qualifications are accurately reflected in their title, and that women's and men's academic titles are used in a parallel fashion | Albert Einstein and Mrs Mead; Ms Clark and John Howard; Judy Smith and Dr Nguyen | Partially |
| 3.2.9. Use of Ms, Mrs, Miss, Mr | The use of 'Ms' is recommended for all women when the parallel 'Mr' is applicable, and 'Ms' should be used when a woman's title of preference is unknown | | ✓ |

continued

| Rule | Description | Examples | |
|---|---|---|---|
| 3.2.10. Avoid patronising expressions | Use the words 'man'/'woman', 'girl'/'boy', 'gentleman'/'lady' in a parallel manner | The girls in the office; Ladies; My girl will take care of that immediately | ✓ |
| 3.2.12. Avoid sexist descriptions | Avoid the use of stereotyped generalisations about men's and women's characters and patterns of behaviour | Strong men and domineering women; assertive men and aggressive women; angry men and hysterical women | ✓ |
| 4.1.1. Derogatory labelling | They are still used, and should be avoided. Some acceptable alternatives for such labels are 'person with Down's Syndrome', 'person with an intellectual disability' | Cripple; mongoloid; deaf and dumb; retarded | ✓ |
| 4.1.2. Depersonalising or impersonal reference | Often people with a disability are referred to collectively as the disabled, the handicapped, the mentally retarded, the blind, the deaf, or paraplegics, spastics, epileptics etc. | The disabled; the handicapped; disabled people; the physically handicapped; a paraplegic; paraplegics; an epileptic; the deaf | ✓ |
| 4.1.3. Stereotyping | Never use the terms 'victim' or 'sufferer' to refer to a person who has or has had an illness, disease or disability | Victim of AIDS; AIDS sufferer; polio victim | ✓ |

## References


Ahmed, S. (2007). The language of diversity. *Ethnic and Racial Studies*, *30*(2), 235–256.

Alfonseca, E., Garrido, G., Delort, J. Y., & Peñas, A. (2013). Whad: Wikipedia historical attributes data: Historical structured data extraction and vandalism detection from the wikipedia edit history. *Language Resources and Evaluation*, *47*(4), 1163–1190.

Augoustinos, M., Tuffin, K., & Every, D. (2005). New racism, meritocracy and individualism: Constraining affirmative action in education. *Discourse and Society*, *16*(3), 315–340.

Aussenac-Gilles, N., & Sörgel, D. (2005). Text analysis for ontology and terminology engineering. *Applied Ontology*, *1*(1), 35–46.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Sebastopol, CA: O'Reilly Media, Inc.

Brading, J., & Curtis, J. (2000). *Disability discrimination: A practical guide to the new law*. London: Kogan Page Series.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on applied natural language processing, association for computational linguistics, Stroudsburg, PA, USA, ANLC '92*, pp. 152–155. doi:10.3115/974499.974526.

Buitelaar, P., Olejnik, D., & Sintek, M. (2004). A protégé plug-in for ontology extraction from text based on linguistic analysis. In *The semantic web: Research and applications*, pp. 31–44. Springer.

Chandrasekaran, B., Josephson, J., & Benjamins, V. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems and Their Applications*, *14*(1), 20–26.

Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2012). Detecting offensive language in social media to protect adolescent online safety. In *Proceedings—2012 ASE/IEEE international conference on privacy, security, risk and trust and 2012 ASE/IEEE international conference on social computing, SocialCom/PASSAT 2012*, pp. 71–80.

Chin, S., Street, W., Srinivasan, P., & Eichmann, D. (2010). Detecting wikipedia vandalism with active learning and statistical language models. In *Proceedings of the 4th workshop on information credibility, WICOW'10*, pp. 3–10.

Cimiano, P., McCrae, J., & Buitelaar, P. (2016). *Lexicon model for ontologies: Community report*. https://www.w3.org/2016/05/ontolex/. Accessed 12 July 2016.

Claude, R., & Weston, B. (1992). *Human rights in the world community: Issues and action*. Pennsylvania: University of Pennsylvania Press.

Colker, R., & Milani, A. (2012). *The law of disability discrimination handbook: Statutes and regulatory guidance*. New York, NY: LexisNexis.

Dance, F. (1970). The concept of communication. *Journal of Communication*, *20*(2), 201–210.

Drummond, N., Rector, A., Stevens, R., Moulton, G., Horridge, M., Wang, H., & Seidenberg, J. (2006). Putting owl in order: Patterns for sequences in owl. In *OWLED*.

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Computing semantic relatedness using wikipedia-based explicit semantic analysis*. pp. 1606–1611.

Gangemi, A., Navigli, R., & Velardi, P. (2003). The ontowordnet project: Extension and axiomatization of conceptual relations in wordnet. In *The OntoWordNet project: Extension and axiomatization of conceptual relations in WordNet*, Vol. 2888, pp. 820–838. Springer.

Garla, V., & Brandt, C. (2012). Ontology-guided feature engineering for clinical text classification. *Journal of Biomedical Informatics*, *45*(5), 992–998.

Hayes, P. J., & Patel-Schneide, P. F. (2014). Rdf 1.1 semantics. https://www.w3.org/TR/rdf11-mt/. Accessed 18 March 2016.

Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on computational linguistics-Volume 2, Association for Computational Linguistics*, pp. 539–545.

Hellmann, S., Lehmann, J., Auer, S., & Brümmer, M. (2013). Integrating NLP using linked data. In *International semantic web conference*, pp. 98–113. Springer.

Horrocks, I. (2008). Ontologies and the semantic web. *Communications of the ACM*, *51*(12), 58–67.

Horrocks, I., Patel-Schneider, P., & Van Harmelen, F. (2003). From SHIQ and RDF to OWL: The making of a web ontology language. *Web Semantics*, *1*(1), 7–26.

Hotho, A., Maedche, A., & Staab, S. (2002). Ontology-based text document clustering. *KI*, *16*(4), 48–54.

Isaac, A., & Summers, E. (2009). *Skos simple knowledge organization system primer*. w3c recommendation. Technical Report, World Wide Web Consortium (W3C).

Kasper, W., & Vela, M. (2012). Sentiment analysis for hotel reviews. *Speech Technology*, *4*(2), 96–109.

Knijff, J., Frasincar, F., & Hogenboom, F. (2013). Domain taxonomy learning from text: The subsumption method versus hierarchical clustering. *Data & Knowledge Engineering*, *83*, 54–69. doi:10.1016/j.datak.2012.10.002.

Kohler, J., Philippi, S., Specht, M., & Ruegg, A. (2006). Ontology based text indexing and querying for the semantic web. *Knowledge-Based Systems*, *19*(8), 744–754.

Kontopoulos, E., Berberidis, C., Dergiades, T., & Bassiliades, N. (2013). Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*, *40*(10), 4065–4074.

Kontostathis, A., Edwards, L., & Leatherman, A. (2009). Chatcoder: Toward the tracking and categorization of internet predators. In *Society for industrial and applied mathematics—9th SIAM international conference on data mining 2009, Proceedings in applied mathematics*, Vol 3. pp. 1327–1334.

Kubota, R., & Lin, A. (2010). *Race, culture, and identities in second language education: Exploring critically engaged practice*. New York: Taylor & Francis.

Li, C., Yang, J., & Park, S. (2012). Text categorization algorithms using semantic approaches, corpus-based thesaurus and wordnet. *Expert Systems with Applications*, *39*(1), 765–772.

Litosseliti, L. (2014). *Gender and language theory and practice*. New York: Taylor & Francis.

Loenen, T., & Rodrigues, P. (1999). *Non-discrimination law: Comparative perspectives*. Alphen aan den Rijn: Kluwer Law International.

Luo, Q., Chen, E., & Xiong, H. (2011). A semantic term weighting scheme for text categorization. *Expert Systems with Applications*, *38*(10), 12,708–12,716.

Machhour, H., & Kassou, I. (2013). Improving text categorization: A fully automated ontology based approach. In *2013 Third international conference on communications and information technology (ICCIT)*, IEEE, pp. 67–72.

Maedche, A., & Staab, S. (2001). Ontology learning for the semantic web. *IEEE Intelligent Systems and Their Applications*, *16*(2), 72–79.

McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gómez-Pérez, A., et al. (2012). Interchanging lexical resources on the semantic web. *Language Resources and Evaluation*, *46*(4), 701–719.

Mowbray, J. (2012). *Linguistic justice: International law and language policy*. Oxford: OUP.

ODP. (2010). Owl list pattern. http://ontologydesignpatterns.org/wiki/Submissions:List. Accessed 18 May 2016.

Orelus, P. (2011). *Rethinking race, class, language, and gender: A dialogue with noam chomsky and other leading scholars*. Lanham, MD: Rowman & Littlefield Publishers.

Salguero, A., & Espinilla, M. (2016). *Description logic class expression learning applied to sentiment analysis*. Cham: Springer. doi:10.1007/978-3-319-30319-2_5.

Santorini, B. (1990). *Part-of-speech tagging guidelines for the penn treebank project (3rd revision)*. Technical Report, University of Pennsylvania.

Schiek, D., & Lawson, A. (2011). *European union non-discrimination law and intersectionality: Investigating the triangle of racial, gender and disability discrimination*. Farnham: Ashgate.

Shuy, R. W. (2007). *Fighting over words: Language and civil law cases: Language and civil law cases*. Oxford: Oxford University Press.

Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics*, *5*(2), 51–53.

Tablan, V., Bontcheva, K., Roberts, I., & Cunningham, H. (2015). Mímir: An open-source semantic search framework for interactive information seeking and discovery. *Web Semantics: Science, Services and Agents on the World Wide Web*, *30*, 52–68. doi:10.1016/j.websem.2014.10.002 http://www.sciencedirect.com/science/article/pii/S1570826814001036, semantic Search.

Talbot, M. (2010). *Language and gender*. New York: Wiley.

Tontti, J. (2004). *Right and prejudice: Prolegomena to a hermeneutical philosophy of law*. Farnham: Ashgate.

University of Newcastle. (2006). *Inclusive language policy 000797*. http://www.newcastle.edu.au/policy/000797.html.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, *11*(2), 93–136.

Uschold, M., Gruninger, M., et al. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, *11*(2), 93–136.

Wang, P., Hu, H. J. J. Z., & Chen, Z. (2009). Using wikipedia knowledge to improve text classification. *Knowledge and Information Systems*, *19*(3), 265–281.

Wei, T., Lu, Y., Chang, H., Zhou, Q., & Bao, X. (2015). A semantic approach for text clustering using wordnet and lexical chains. *Expert Systems with Applications*, *42*(4), 2264–2275. doi: 10.1016/j.eswa.2014.10.023.

Weller, P., Purdam, K., Ghanea, N., & Cheruvallil-Contractor, S. (2013). *Religion or belief, discrimination and equality: britain in global contexts*. London: Bloomsbury Publishing.

Xu, H., Zhang, F., & Wang, W. (2015). Implicit feature identification in chinese reviews using explicit topic mining model. *Knowledge-Based Systems*, *76*, 166–175. doi:10.1016/j.knosys.2014.12.012.

Yates, S. (2001). Gender, language and CMC for education. *Learning and Instruction*, *11*(1), 21–34.

Zhang, F., Ma, Z., & Li, W. (2015). Storing owl ontologies in object-oriented databases. *Knowledge-Based Systems*, *76*, 240–255. doi:10.1016/j.knosys.2014.12.020.