Modeling Natural Language Policies into Controlled Natural Language: A Twitter Case Study

Irfan Khan Tanoli

Computer Science Department
University of Beira Interior
Covilha, Portugal, 6201-001
Email: irfan.khan.tanoli@ubi.pt

Sebastião Pais

Computer Science Department
NOVA LINCS and UBI
Covilha, Portugal, 6201-001
Email: sebastiao@di.ubi.pt

Abstract—**Social network providers usually describe the terms of data storage, usage, and sharing, by adopting natural languages. To automatically evaluate such terms of use, to understand, analyse, and enforce rights and obligations over the user's data, it is of uttermost importance to translate them in a machine-readable format. Natural Languages (NLs) are the most prominent form of knowledge representation for humans. However, due to NLs complexities, it is quite burdensome to process their sentences by machines in a seamless and standardised way. Controlled Natural Languages (CNLs) are subsets of NLs that are obtained by restricting the grammar and vocabulary, to minimize - or even eliminate - ambiguity and complexity of NL. These languages hold two major characteristics: they look informal and easy to read and write by humans, quite like natural languages, but they can be easily transited into machine-readable forms. In this paper, we study some policy-oriented CNLs. We adopt them as source languages for translating sample Twitter policies. Then, we assess the value of the different languages, according to the difficulties of the translation, its readability, and other compelling properties to find which CNL is more suitable for NL translation.**

Keywords–Natural Language; Controlled Natural Languages; Social Networks; Natural Language Processing; Data Policies.

I. INTRODUCTION

Social Networks (SNs) have a great impact on our everyday life. Users increasingly rely on SNs to share their opinions, plan activities, exchange information, and establish social relationships. SNs interactions usually require the exchange of users' data for a variety of purposes, including the provisioning of services. The collection, usage, and sharing of user's data is usually regulated by social networks (e.g., Facebook data [1], Twitter privacy policies [2], Google privacy policies [3]). Usually publish in English NL, the policies describe the terms and condition under which the provider will manage the data in terms of e.g., authorised, obliged, or denied. Although the use of Natural Language (NL) enables end users to read and understand the authorised (or obliged, or denied) operations on their data, a key issue relies on the fact that NLs are not machine readable, and automatic controls on how the data are actually going to be used and processed by the entities that operate on them is not feasible.

In particular, NLs cannot be used as the input language for a policy-based software infrastructure to be used for policy management. In fact, both automated policy analysis (the process to assure the lack of conflicting data policies, see, e.g., [4] [5]) and policy enforcement (the actual application of the data policies, whenever a data access request takes place) require inputs in a machine readable form, like, e.g., the *de facto* standard eXtensible Access Control Markup Language (XACML) [6] [7]. With the aim of moving in the direction of managing and enforcing access policies automatically, in this paper we consider a selection of different machine-oriented, English-based CNLs, originally designed within different contexts, and we investigate their effectiveness in expressing data policies as specified on a popular SN site.

CNLs are a subset of NLs, specifically conceived to make machine processing simpler. A CNL is, in essence, a developed language that is based on NL, but it is more restrictive in terms of lexicon, syntax, semantics, while at the same time retaining most of its natural properties [8]. CNLs have more contrived representation, in terms of grammar and vocabulary, and they thus reduce the ambiguity and complexity of a complete language [9], e.g., English, Spanish, French, Swedish, Mandarin, etc. [10]. CNLs have been proved to be effective in mitigating linguistic ambiguity challenges, as they can easily be translated into a formal language such as, first-order logic or different version of description logic, automatically and mostly deterministically [9].

CNLs can be roughly classified into two broad classes: human-oriented and machine-oriented. Human-oriented CNLs mostly used for improvement of technical documentation readability and comprehensibility. Machine-oriented CNLs are purposely dedicated to refine the translation of complex and technical documents [11], for knowledge presentation or processing [12], and for the Semantic Web [13]. Machine-oriented CNLs can also support translation of large texts, e.g., in English, into first-order logic, to automatically map their expressiveness into a small subset of expressions [13]. CNLs can be developed for specific scenarios and application domains [9]. For example, the Attempto Controlled English (ACE) [12] [14] has been designed with an expressive

 ISBN: 978-1-61208-804-4

knowledge representation that is easy to learn, read and write for domain experts.

The variety of CNLs attributes suggests that it is difficult to identify their general properties. First, CNLs are defined for different areas, (e.g., academia and industry), and for different fields, (e.g., computer science, mathematics, engineering, linguistics, etc.) Secondly, even if CNLs usually share common properties, there can be either CNLs that are inherently ambiguous, or precise as formal logic. Some are quite natural, others are closer to programming languages or to logic-based formalisms, or just defined with simple grammar rules, others are more complex and their syntax and semantics are not easy to define and/or understand [8]. Due to such variations, it is difficult to define fundamental properties to be used for comparing different CNLs.

Here, we consider samples of real Twitter data policies for translating from their original form in natural language to each of the selected controlled languages. Google [3], Facebook [1] and Twitter [2] data policies express the same actions like how user's data is regulated. The reason for choosing Twitter as a sample case study for translation is arbitrary, although Facebook and Google policies can also be translated in the same way. The translations are evaluated with respect to key properties defined in the so-called *Precision, Expressiveness, Naturalness, Simplicity* (PENS) classification scheme [8], having one new property, namely *policy enforcement*. The evaluation will help researchers to choose the most appropriate CNL and to automatically process the terms and conditions under which user's data are accessed, stored, and used for machine readability. The main contribution of this study is to provide an understanding related to CNLs and the need for NL translation into CNL for machine understandability. This study help us finding a certain CNL for NL policy translation. After our finding, we are motivated for development of an automated system that can translates CNL into NL.

The rest of the paper is organized as follows: Section II presents an overview of the three CNLs. Section III describes the key properties of the PENS scheme with a new general one. Section IV introduces some sample Twitter policies and their translations into each of the targeted CNLs. By relying on the translations, Section V presents an assessment and comparison of the considered CNLs. The final section VI outlines directions for future work and draws the conclusions.

II. Controlled Natural Languages

CNLs are general-purpose languages designed to facilitate domain experts in expressively representing knowledge. On the one side, they are easy to learn, write and read, but, on the other side, they are meant to be fully machine-readable (or, at least, designed in a way that makes possible their automatic translation into a machine-readable language). In this section, we consider three different machine-oriented policy-based languages. For our study, we include both general-purpose and domain-specific controlled languages, originally targeted at different contexts, e.g., knowledge representation, and policy authoring and enforcement.

Throughout the section, we present three sample policies in natural language and we translate them in each of the three languages. The sample policies we will consider are the following:

- **E1:** *User can log into system with valid Id and Password.*
- **E2:** *Bob must send documents to Alex, when Alex requests to Bob.*
- **E3:** *Ryan cannot share Paulo's data, if Paulo disallows Ryan.*

A. Attempto Controlled English

Attempto Controlled English (ACE) [12] [14] is a CNL developed for an automatic and unambiguous translation into a first-order logic. It was initially designed as a specification language, but the language has been improved over the years in various ways, gradually shifting towards knowledge representation and applications for the Semantic Web [8]. ACE has a few small set of construction and set of interpretation rules. The former explains its syntax and the latter makes the constructs clear which are vague in full English. ACE has a vocabulary which consists of some function words (conjunctions, pronouns), fixed phrases (there is), and content words (nouns, verbs, adjectives). Definitive Clause Grammar (DCG) is used to write grammars upon which the language processor relies. DCGs are equipped with certain structures that convert declarative and interrogative sentences into first-order logic. Once the discourse representation structure is created only then can anaphoric references be resolved. ACE also provides support for active and passive words, subject and object relative clauses [9].

ACE is intended for researchers who wish to use formal notation and formal methods even though they are not familiar or expert with them [15]. Notable features of this controlled language include the capability to express complex noun phrases, plurals, anaphoric references, subordinated clauses, modality, and questions. In ACE, the previously introduced sample policies can be expressed as:

- **E1:** `A user has a valid ID and PASSWORD to log into system and system validates ID and PASSWORD.`
- **E2:** `If Alex requests Bob THEN Bob must send documents to Alex.`
- **E3:** `If Ryan disallows Paulo then Paulo cannot share Ryan's data.`

There exists other CNLs similar to ACE for knowledge representation: as an example, Processable English (PENG) [16], Computer Processable Language (CPL) [17], Common Logic Controlled English (CLCE) [18], and Formalized English [19]. The comparison amongst this group of languages has been already presented [11]. Here, we decided to consider ACE because of its generality and its features, that render it more expressive, both syntactically and semantically [12].

B. Protune

The Protune (Provisional Trust Negotiation) policy language [4] is based on logic programming and, is

designed for policy evaluation, enforcement, and negotiation [5]. The language is based on standard logic rules of the form $A \leftarrow L_1,....,L_i$ where A is a standard logical atom (called the head of the rule) and $L_1,....,L_i$ (the body of the rule) are literals (that is L_i equals B_i or $\neg B_i$ for some logical atom B).

The format of Protune policy rules is as follows:

```
allow(action) ← condition_1....condition_n
  condition ← condition_1....condition_n
```

An action is allowed if all the conditions are satisfied. The rendering in Protune of the three sample policies is the following:

- **E1:** `allow(loginsystem) ← user(userid=U, password=P):'valid'`
- **E2:** `allow (send(Bob,Alex,documents)) ← request(Alex,Bob)`
- **E3:** `allow (share#Not(Ryan,Paulo,Data)) ← disallow(Paulo,Ryan,Data)`

C. Logic Based Policy Analysis Framework

A logic-based policy analysis language for policy specifications is presented in [20], which comes with a policy analyser providing also diagnostic information about detected conflicts, separation of duty, coverage gaps, behavioural simulation and policy comparison. vLBPAF is developed using the abductive constraint logic programming (ACLP) system as basis for algorithm analysis, and on the Event calculus [21] to represents how events and actions happening that affect states of the system, leading to circumstances in which a given policy rule is applicable and the information is an output of the analysis. The language uses a number of sorted first-order logic predicates, and discriminates between policy language and domain description language. The policy language representation $\mathcal{L}^{\pi}$ consists of sorts for subjects *Sub*, actions *Act*, and targets *Tar*, together with a sort for time *T*, represented using the non-negative reals.

The three $\mathcal{L}^{\pi}$ predicates, referred as 'regulatory predicates', are as follows:

- **Input Regulatory:**
  `req(Sub,Tar,Act,T)`
- **Output Regulatory:**
  `do(Sub,Tar,Act,T)`
  `deny(Sub,Tar,Act,T)`
- **State Regulatory:**
  `permitted(Sub,Tar,Act,T)`
  `denied(Sub,Tar,Act,T),`
  `obl(Sub,Tar,Act,T_s,T_e,T)`
  `fulfilled(Sub,Tar,Act,T_s,T_e,T)`
  `violate(Sub, Tar, Act, T_s, T_e, T)`
  `cease_obl(Sub, Tar, Act, T_init, T_s, T_e,T)`

The input regulatory predicate represents a request for *Sub* to perform *Act* on *Tar*, at time *T*. The output regulatory predicates indicate whether an *Act* is permitted or denied, for *Sub* to *Tar*, at time *T*. The state regulatory predicates indicate different situations concerned with the permitted and denied actions, the fact that an obligation exists, the fact that obligation has been actually fulfilled, violated, or expired. *T* indicates the actual time, while the pair T_s, T_e represent the interval time for the existence of an obligation. As a matter of fact, there exist translations of LPBAF to Ponder [22] and XACML. Both the target languages are enforceable, meaning, they serve as input to a standard policy enforcement infrastructure *a la* XACML. Again, let us see how the three sample properties are rendered in LPBAF:

- **E1:** The action login is permitted by the user 'U' on the system 'S', at time 'T', whenever at time 'T' the user has a valid Id and Password P (holdsAt is based on Event Calculus):
  `permitted(U,S,login,T) ←`
  `holdsAt(U,(Id,P),valid,T)`
- **E2:** In the language notation, 'B' (Bob) is obliged to send to 'A' (Alex) the documents 'D', at time 'T', when 'A' requests to 'B', at time 'T'.
  `obl(B,A,D,send,T) ← do(A,B,request,T)`
- **E3:** Considering Ryan 'R' cannot share Paulo 'P' Data 'D'. The prohibition is enabled if 'P' prohibits 'R' to share it. 'T', a variable rather than a fixed time, signals the beginning of the prohibition.
  `denied(R,P,D,share,T) ←`
  `do(P,R,disallow,T)`

III. Properties for Controlled Natural Languages (CNLs)

A well-established classification scheme, known as *Precision, Expressiveness, Naturalness, Simplicity (PENS)*, has been presented in [8] to support CNL comparison and classification.

A. The PENS Classification Scheme

A standard classification scheme is the better approach for controlled natural languages analysis to determine whether a language fulfills specific characteristics. The Precision, Expressiveness, Naturalness, Simplicity (PENS) scheme [8] was defined following the intuition that CNLs place themselves in between natural and formal languages. In general, CNLs are quite structured and constrained (thus, closer to pure formal languages), still, their syntax is close to natural terms. Furthermore, to establish a general, but, at the same time, restricted classification, the PENS scheme considers English as a natural language and propositional logic as a formal language.

To develop a base classification scheme, it is essential to put the properties under a few dimensions, to avoid as much as possible dependence between each other [8]. The PENS classification scheme considers only four properties *Precision, Expressiveness, Naturalness, Simplicity*, to condense under those umbrellas, the highest number of possible characteristics. For example, attributes like ambiguity in the text, formal definition of language, and capability to transform the language into a propositional logic can be merged under the Precision dimension. Natural writing, natural feeling and understanding of the language can be put under the Naturalness dimension. Instead, Simplicity measures the non-complexity of the language. The expressiveness of

 ISBN: 978-1-61208-804-4

a language is a measure of the variety of lexical and grammatical constructions, it allows (irrespective of the reader).

In the following, we will consider such four properties as the standard base for our comparison, plus one more property *Policy enforcement*, which is discussed later in this section. Each of the PENS dimensions is measured through five classes, ranging over the interval 1, ..., 5. Each of the five classes presents a one-dimensional area between the two extremes, i.e., English at one end and propositional logic on the other one. The decision to assign a language to one of the five classes, for each dimension, is left arbitrary. Considering Simplicity and Precision, English is at the bottom, i.e., S^1 and P^1, while propositional logic is at the top, S^5 and P^5. Conversely, for Expressiveness and Naturalness, English is at the top: E^5 and N^5 while propositional logic is at the bottom: E^1 and N^1The complete details are available in [8]. The five classes for each dimension are described in a vast scope and cover a wide range of CNLs. Therefore, to make a simple, but effective comparison among the languages as described in (Sect. II), we select only one class for each dimension (usually, a class in the middle).

1) Precision: Precision is referred to as the degree to which the meaning of a text can be directly understood and recovered from its textual form in a particular language, i.e., the sequence of linguistic symbols [8]. The ambiguity in the meaning, predictability, and formality of the definition can be combined with precision. Formal logic languages are highly precise because the meaning of the text is strictly defined based on the possible sequences of the symbols of the language, as compared to NLs which are, according to the property definition, imprecise and ambiguous.

The precision classes are defined as: Imprecise languages, Less imprecise languages, Reliably interpretable languages, Deterministically interpretable languages, Languages with fixed semantics., we select **'Deterministically Interpretable Languages (DIL)'** as the reference class: this class includes languages that are entirely formal at the *syntactic* level. Texts in this language can be deterministically translated into a logical representation that defines the meaning of sentences. However, any sensitive deduction may require additional background axioms, external or heuristic resources [8].

2) Expressiveness: Expressiveness is related to the range of propositions that a language is capable of expressing. For example, language 'Y' is more expressive than language 'Z' if 'Y' can describe all that 'Z' can, but 'Z' cannot do the same w.r.t. 'Y'. This relationship does not necessarily induce a total order. For example, given two languages, it might be that none of them is more expressive than the other one. This makes it hard, or even unfeasible, to objectively rank in a linear order a set of languages, in terms of expressiveness [8].

PENS consider the following characteristics of expressiveness:

1) universal quantification over individuals, i.e., the presence in the language syntax of the logical predicate ∀, 'given any' or 'for all'.
2) relations of arity greater than one, i.e., languages which functions/predicates are taking as input more than one argument.
3) general rule structures, e.g., if-then-else conditions.
4) negation (failure or strong negation).
5) second-order (extension of first-order logic) universal quantification over concepts and relations [23].

By considering the above characteristics, it is possible to categorize languages according to five different classes of expressiveness: inexpressive languages, languages with low expressiveness, languages with medium expressiveness, languages with high expressiveness and languages with maximal expressiveness, we focus on **'Languages with Medium Expressiveness (LwME)'**, i.e., languages with all the characteristics of expressiveness as above, except second-order universal quantification.

3) Naturalness: The dimension of naturalness defines how a language is 'natural' in terms of reading and understanding from the user standpoint. Linguistic properties such as modification of grammar, comprehensibility, and natural reading and writing can be considered elements of naturalness. CNLs retains most of the natural properties of native languages, so that native language users can, quite effortlessly, understand texts without the need of language experts. The five naturalness classes are: unnatural languages, languages with dominant unnatural elements, languages with dominant natural elements, languages with natural sentences, languages with natural texts. This study considers **'Languages with Dominant Natural Elements (LwDNE)'** as point of reference, this study considers *'Languages with Dominant Natural Elements (LwDNE)'* as a point of reference.

With these types of languages, natural elements of languages dominate unnatural elements, and the overall grammar structure corresponds to the grammar of the natural language. However, due to the rest of natural elements or combination of unnatural elements, these languages cannot be considered valid natural sentences. Natural language speakers cannot easily recognize the sentences statements and cannot understand their essence without any guidance or instructions but still intuitively understand the language to a substantial degree [8].

4) Simplicity: Simplicity is consider as how simple (resp., complex) is to describe the language accurately and comprehensively, covering syntax and semantics. These 'exact and comprehensive descriptions should define all syntactic and semantic properties of the language using accepted grammar notations to define the syntax and accepted mathematical or logical notations to define the semantics. Concerning the PENS classification scheme, the indicator of simplicity is the number of natural language pages needed to describe the language accurately and comprehensively, consisting in the definition of all the syntactic and semantic properties of the language. Page counting should be done considering a single-column format, with a maximum of 700 words per page. The language descriptions do not require to include vocabularies [8].

 ISBN: 978-1-61208-804-4

From the following five properties of simplicity, i.e., very complex languages, languages without exhaustive descriptions, languages with lengthy descriptions, languages with short descriptions and languages with very short descriptions, we consider **'Languages with Short Descriptions (LwSD)'** as the term of comparison: a language considered to be simple enough to be described in more than a single page but less than ten pages.

B. Policy Enforcement

A standard architecture for the application (technically, 'enforcement') of privacy policies is as follows. Consider a generic subject 'S' that tries to access the object 'O' (a medical report, a picture published on a social network, etc.) to, e.g., modify or delete it or share it with third parties. This sketched architecture is adopted by the most common and tested authorization and control systems, such as the one implemented in the authorization infrastructure associated with XACML [6] [7]. We will thus consider a further property, Policy Enforcement (PE), taking into accounts if the CNLs under investigation are enforceable, or not. In other words, we will consider if they serve as input to standard tools for policy enforcement.

IV. TANSLATION OF TWITTER POLICIES

In this section, we consider real Twitter policies and present their translation into each of the three selected CNLs. The outcome will be evaluated in Section V, to assess the relative merits of the considered CNLs with respect to Precision, Expressiveness, Naturalness, Simplicity (PENS), and amenability to Policy Enforcement (PE).

A. Twitter Data Policies

The Twitter Data Policies [2], describe the kind of information collected by the social network and how such information is used and shared. Hereafter, we consider the following sample policies.

- **Contact Information and Address Books:**
  P1: You can choose to upload and sync your address book on Twitter so that we can help you find and connect with people[...].
- **Twitter for Web Data:**
  P2: When you view our content on third-party websites that integrate Twitter content such as embedded timelines or Tweet buttons, we may receive Log Data that includes the web page you visited.
- **Developers**
  P3: If you access our APIs or developer portal, we process your personal data to provide our services..
- **Object, Restrict, or Withdraw Consent**
  P4: When you are logged into your Twitter account, you can manage your privacy settings and other account features here at any time.
- **Accessing or Rectifying Your Personal Data**
  P5: If you have registered an account on Twitter, we provide you with tools and account settings [...].

B. From natural to controlled natural languages

Below, we show examples of translations of the Twitter policies listed above to the CNLs described in (Sect. II). Here, we consider P1, P2, P3, P4, P5.

1) Attempto Controlled English:

P1 in ACE:

```
IF You can choose to upload and sync your
address book on Twitter THEN we can help you
find and connect with people.
```

P2 in ACE:

```
IF you view our content on third-party
websites that integrate Twitter content such
as embedded timelines or Tweet buttons THEN we
may receive log data that includes the web
page you visited.
```

P3 in ACE:

```
IF you access our APIs or developer portal
THEN we process your personal data to provide
our services.
```

P4 in ACE:

```
IF you are logged into your Twitter account
THEN you can manage your privacy settings and
other account features here at any time.
```

P5 in ACE:

```
IF you have registered an account on Twitter
THEN we provide you with tools and account
settings.
```

2) Protune (PROvisional TrUst NEgotiation):

P1 in Protune:

```
allow (help(we,you,(FindandConnect(people))))
← ChoosetoUpload (you,address book,Twitter),
  ChoosetoSync (you,address book,Twitter)
```

P2 in Protune:

```
allow (receive(We,LogData) ←
  visit (you,web page), view
(our,content,third-party website),
  integrate(twitter,content),
content:timeline,tweet buttons.
```

P3 in Protune:

```
allow (process(your,personal
data,(provide(our,services))) ←
access (you,our (API developer portal))
```

P4 in Protune:

```
allow (manage#atanyTime (your,privacy
                settings,
other account features))← log (you,twitter
                account)
```

P5 in Protune:

```
allow (provide(we,you,tools,account
settings))← register (you,twitter account)
```

3) Logic Based Policy Analysis Framework:

P1 in LBPAF:

If you 'Y' choose to upload and sync Address Book 'AB' on Twitter 'TW' THEN we 'W' can help 'Y' find and connect with people 'P' in Time 'T'. 'T' in the head of the rule is a variable rather than a fixed time and it has been inserted since required by the syntax of LBPAF.

```
permitted(W,Y,help(P,find,connect,T) ←
   do(Y,C,AB,TW,ChoosetoUpload,T), do
            (Y,AB,TW,sync,T)
```

P2 in LBPAF:

If you 'Y' view our content 'C' on third-party website 'TPW' that integrate Twitter content 'TC' such as embedded timelines 'ET' or Twitter buttons 'TB' THEN we 'W' may receive that Log Data 'LD' that inluded page 'P', 'Y' visited in Time 'T'. 'happens' is based on Event Calculus.

```
        permitted(W,LD,receive,T) ←
do(Y,TC,TPW,view,T),holdAt(TC,(ET,TB,T),
   integrate,T),happens(Y,P,visited,T)
```

P3 in LBPAF:

```
permitted(W,YP,D,process,(TW,provide),T) ←
              do(Y,TA,access,T)
```

P4 in LBPAF:

If you 'Y' logged' into your Twitter account 'TA', you 'Y' can manage your privacy settings 'PS' and other account feature 'OAF' in Time 'T'.

```
permitted(Y,PS,manage,T) ← do(Y,TA,log,T)
```

P5 into LBPAF:

If you 'Y' register an account 'A' on Twitter 'T' THEN We 'W' provide you'Y' with tools 'TO' and account settings 'AS' in Time 'T'.

```
permitted(W,Y,TO,AS,provide,T) ←
       do(Y,A,T,register,T)
```

V. EVALUATION

In this section, we consider the three languages discussed in Section II and we evaluate to which degree they fulfil the properties introduced in Section III based on the translation presented in Section IV.

A. ACE

ACE (Sect. II-A) is a precise language, according to the definition of *precision* (Sect. III-A1) and, in particular, it can be classified as a *Deterministically Interpretable Language* (completely formal at syntactic level). In terms of *expressiveness* (Sect. III-A2), ACE can be classified as a *Language with Medium Expressiveness*. As noticed in [8], it has general rule structures, negation, arity relation greater than one and universal quantification over individuals [12]. In terms of *naturalness*, ACE cannot be considered as a *Language with Dominant Natural Elements* as discussed in [8]. The Twitter policies in Sect. IV-B1 can be easily understood by a general audience without external guidance. For *simplicity*, authors in [8] define ACE as *Language with Lengthy Descriptions* [12], [24]. ACE is also not a *policy-enforceable* language (Sect. III-B, being not associated to any policy enforcement architecture [12].

TABLE I. COMPARISON OF CONTROLLED NATURAL LANGUAGES

	ACE	Protune	LBPAF
Precision (DLI)	Yes	Yes	Yes
Expressiveness (LwME)	Yes	Yes	Yes
Naturalness (LwDNE)	No	No	No
Simplicity (LwSD)	No	Yes	Yes
Policy Enforcement	No	Yes	Yes

B. Protune

Being equipped with a formal syntax, Protune (Sect. IV-B2) holds the *precision* property, with degree *Deterministically Interpretable Languages*. Protune meets all the four features needed for being classified as a *Language with Medium Expressiveness* (Sect. III-A2): general rules structure, negation, universal quantification over individuals, and relations of arity grater than one [25]. Protune features a mixture of natural and unnatural terms and its grammar structure does not correspond to that of a natural language (Sect. IV-B2).

Proper guidance is needed to adopt Protune; users fail to intuitively understand the respective statements [8]. Therefore, our opinion is that it cannot be classified as a *Language with dominant natural elements* (Sect. III-A3). Protune is described with exact and comprehensive syntax and semantics and the language description is more than a single page but less than 10 pages [25]; hence, it can be categorized as a *Language with Short Descriptions* (Sect. III-A4). Finally, Protune supports *policy enforcement* [15] [24].

C. Logic Based Policy Analysis Framework

Logic Based Policy Analysis Framework (LBPAF) is a *precise* language (Sect. III-A1), fully formal and fully specified both at the syntactic and at the semantic level. The language is a *Deterministically Interpretable Language*. LBPAF is an *expressive* language for policy definition, in particular, it enjoys the four properties needed for being a *Language with Medium Expressiveness* [20]. Non expert people need proper guidance for using the language. Moreover, as shown in Sect. IV-B3, the unnatural elements are dominant with respect to the natural ones. Therefore, we cannot classify LBPAF as a *Language with Dominant Natural Elements*. Regarding *simplicity*, the language description is such that it takes more than a single page but less than 10 pages [20]. Therefore, this language can be classified as a *Language with Short Descriptions*. Finally, it can be translated into the enforceable language Ponder [22], fulfilling, even if indirectly, the property of *policy-enforcement*.

D. Summary

Our analysis is summarised in Table I, where rows indicate the policy languages and columns indicate the properties. Intuitively, cells are marked with 'Yes' or 'No', according to whether or not a language satisfies a

 ISBN: 978-1-61208-804-4

certain property. The evaluation shows that Protune and LBPAF fulfils the highest number of properties. The two languages are formal at the syntactic level and have an associated formal semantics; their description is concise, thus fulfilling the simplicity property at level of languages with shorts descriptions; they were not defined with a specific vocabulary associated and all have a policy enforcement infrastructure associated. Protune and LBPAF enjoy the property of medium expressiveness, and none of the language appears to be a language with dominant natural elements.

VI. Conclusions

In this study, we considered three Controlled Natural Languages and we evaluated them according to a set of standard properties defined in the literature. The evaluation is carried out based on the translation of a Twitter policies into the analysed CNLs. Findings are that, according to the PENS scheme, all languages are formal at the syntactic level (remarkably, all but ACE also have a precise semantics associated). The three languages feature different degrees of expressiveness (in terms of expressible logical operators and functions), presence of natural elements, and simplicity of their descriptions. Finally, two out of three i.e., Protune and LBPAF serve as input to a standard policy enforcement infrastructure *a la* XACML.

Notably, each of the investigated languages is capable of expressing data privacy policies. Aiming at choosing a CNL as the target language to automatically translate NL social network(s) data policies, the outcome of our evaluation helps us towards Protune and LBPAF. However, both languages are rigorous at the syntactic and semantics level, expressive enough, and they do not need a huge effort in terms of learning of use. A notable remark is that they come with devoted toolkits for policy authoring, analysis and enforcement. For future work, we aim at designing a CNL (or adapting an existing one, possibly among the ones investigated in this work) easily understandable and sufficiently expressive to be used directly by the managers of social network sites, to describe the use they make of the data that users provide them.

Acknowledgment

This work was supported by National Founding from the FCT- Fundacão para a Ciência e a Tecnologia, through the MOVES Project - PTDC/EEI-AUT/28918/2017 and by operation Centro-01-0145-FEDER-000019-C4 - Centro de Competências em Cloud Computing, co-financed by the ERDF through the Centro 2020, in the scope of the Sistema de Apoio à Investigação Científica e Tecnológica.

References

[1] "Data Policy," 2020, URL: https://www.facebook.com/full_data_use_policy [accessed: 2020-09-15].

[2] "Twitter Privay Policy," 2020, URL: https://twitter.com/en/privacy [accessed: 2020-09-15].

[3] "Google Privacy and Terms," 2020, URL: https://policies.google.com/privacy [accessed: 2020-09-15].

[4] J. L. De Coi, D. Olmedilla, P. A. Bonatti, and L. Sauro, "Protune: A framework for semantic web policies." in International Semantic Web Conference (Posters & Demos), vol. 401, 2008, p. 128.

[5] P. Bonatti, J. L. De Coi, D. Olmedilla, and L. Sauro, "A rule-based trust negotiation system," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 11, 2010, pp. 1507–1520.

[6] B. Parducci, H. Lockhart, and E. Rissanen, "Extensible access control markup language (xacml) version 3.0," OASIS Standard, 2013, pp. 1–154.

[7] D. Ferraiolo, R. Chandramouli, R. Kuhn, and V. Hu, "Extensible access control markup language (xacml) and next generation access control (ngac)," in Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control, 2016, pp. 13–24.

[8] T. Kuhn, "A survey and classification of controlled natural languages," Computational Linguistics, vol. 40, no. 1, 2014, pp. 121–170.

[9] T. Gao, "Controlled natural languages for knowledge representation and reasoning," in Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016, p. 21.

[10] T. Khun et al, "On controlled natural languages: Properties and prospects," in International Workshop on Controlled Natural Language. Springer, 2009, pp. 281–289.

[11] R. Schwitter, "Controlled natural languages for knowledge representation," in Coling 2010: Posters, 2010, pp. 1113–1121.

[12] N. Fuchs, K. Kaljurand, and T. Kuhn, "Attempto Controlled English for knowledge representation," Reasoning Web, 2008, pp. 104–124.

[13] H. Safwat and B. Davis, "Cnls for the semantic web: a state of the art," Language Resources and Evaluation, vol. 51, no. 1, 2017, pp. 191–220.

[14] N. E. Fuchs, "Understanding texts in attempto controlled english." in CNL, 2018, pp. 75–84.

[15] J. De Coi, P. Kärger, D. Olmedilla, and S. Zerr, "Using natural language policies for privacy control in social platforms." CEUR Workshop Proceedings, ISSN 1613-0073, 2009.

[16] C. White and R. Schwitter, "An update on PENG light," in ALTA, vol. 7, 2009, pp. 80–88.

[17] P. Clark, W. R. Murray, P. Harrison, and J. Thompson, "Naturalness vs. predictability: A key debate in controlled languages," in Controlled Natural Language. Springer, 2009, pp. 65–81.

[18] J. F. Sowa, "Common Logic Controlled English," URL: http://www.jfsowa.com/clce/specs.htm [accessed: 2020-09-15].

[19] P. Martin, "Knowledge representation in CGLF, CGIF, KIF, frame-CG and Formalized-English," in Conceptual Structures: Integration and Interfaces. Springer, 2002, pp. 77–91.

[20] R. Craven et al, "Expressive policy analysis with enhanced system dynamicity," in Symposium on Information, Computer, and Communications Security. ACM, 2009, pp. 239–250.

[21] R. Kowalski and M. Sergot, "A logic-based calculus of events," New Generation Computing, vol. 4, no. 1, Mar 1986, pp. 67–95.

[22] G. Russello, C. Dong, and N. Dulay, "Authorisation and conflict resolution for hierarchical domains," in 8th IEEE International Workshop on Policies for Distributed Systems and Networks, 2007, pp. 201–210.

[23] Stanford Encyclopedia of Philosophy, "Quantifiers and quantification," 2018, URL: https://plato.stanford.edu/entries/quantification/#SecOrdQua[accessed: 2020-09-15].

[24] J. De Coi, N. E. Fuchs, K. Kaljurand, and T. Kuhn, "Controlled English for reasoning on the Semantic Web," in Semantic techniques for the web. Springer, 2009, pp. 276–308.

[25] P. Bonatti and D. Olmedilla, "Driving and monitoring provisional trust negotiation with metapolicies," in Policies for Distributed Systems and Networks. IEEE, 2005, pp. 14–23.

 ISBN: 978-1-61208-804-4