Comparing natural language documents: a DL based approach

Naouel Karam, Michel Schneider
LIMOS, Université Blaise Pascal, France
{karam, schneider}@isima.fr

Abstract

We propose a method to compare semantically two natural language texts. The process is realized in two steps, the first translates the texts into description logics terminologies. The second computes the difference between the terminologies obtained. We show how the best covering problem can be used to compute the difference between two terminologies and propose a method to calculate this difference.

1 Introduction and motivation

This paper deals with the problem of comparing Natural Language (NL) texts. The motivations behind this work come from different applications, like for example document indexation and web sites maintenance, where one needs to compare the documents and characterize their difference in a precise way. To achieve this task, we propose a process in two steps:

1. **the translation step** that aims to formally represent the semantics of the two natural language texts;
2. **the comparison step** that uses an algorithm to compute the difference between the descriptions obtained.

We propose to use description logics (DLs) [1] as a formal representation language to describe the NL semantics. The motivations are twofold: DLs come with well-defined semantics and correct inference algorithms and the formalization of a text in DLs has already been studied [5].

For the translation step we reuse the principles described in [5] that makes the connection between natural language and DLs based on the observation that natural language semantics where formalized by relational algebras [2, 6] and that this later have a link with DLs [4]. The process works as follows: given a text in natural language, first, construct its algebraic representation, then transform the algebraic expressions into DL statements. For a text in natural language we obtain a set of concept definitions (a *terminology*) describing its semantics.

The comparison step consists in comparing the two terminologies obtained. Given two terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$ describing two texts $text1$ and $text2$ respectively, our goal is to find all the extra information contained in $\mathcal{T}_1$ and not in $\mathcal{T}_2$ and vice-versa. In order to characterize the extra information, we need to find the common information

between all the concept definitions occurring in the first terminology and the second terminology. This is done by a mapping ρ that associates each concept A_i in $\mathcal{T}_1$ to a combination of concepts in $\mathcal{T}_2$ that contains as much as possible of common information with A_i and as less as possible of extra information with respect to A_i. $\rho(A_i)$ is then the best cover of A_i using $\mathcal{T}_2$. The problem of discovering the best cover of a concept using a terminology has been formalized in [3].

We describe how the best covering problem can be used to compute the difference and the dissimilarity coefficient between two terminologies.

2 The best covering problem

The notion of best cover was formally defined in [3], it was applied to the dynamic discovery of e-services. The authors characterize the notion of extra information with the help of a difference operation defined in the framework of description logics where the difference is always semantically unique.

We recall the definitions introduced in [3] to formally define the best covering problem. Let $\mathcal{L}$ be a description logic with structural subsumption, $\mathcal{T}$ be an $\mathcal{L}$-terminology, and $Q \not\equiv \bot$ an $\mathcal{L}$-concept description.

Definition 1 *(cover) A cover of a concept Q using $\mathcal{T}$ is a conjunction E of some defined concept names occurring in $\mathcal{T}$ such that: $Q - lcs_{\mathcal{T}}(Q, E) \not\equiv Q$.*
Here $lcs_{\mathcal{T}}(C, D)$ is the least common subsumer of the concepts C and D w.r.t a terminology $\mathcal{T}$.

Definition 2 *(rest and miss) Let Q be an $\mathcal{L}$-concept description and E a cover of Q using $\mathcal{T}$. The rest of Q with respect to E, written $Rest_E(Q)$, is defined as follows: $Rest_E(Q) \doteq Q - lcs_{\mathcal{T}}(Q, E)$.*
The missing information of Q with respect to E, written $Miss_E(Q)$, is defined as follows: $Miss_E(Q) \doteq E - lcs_{\mathcal{T}}(Q, E)$.

Definition 3 *(best cover) A concept description E is called a best cover of Q using a terminology $\mathcal{T}$ iff:*

- *E is a cover of Q using $\mathcal{T}$, and*
- *there doesn't exist a cover E' of Q using $\mathcal{T}$ such that $(|Rest'_E(Q)|, |Miss'_E(Q)|) < (|Rest_E(Q)|, |Miss_E(Q)|)$.*

3 Extracting terminology from text

3.1 Linking DL representation and relation algebras

The semantics of DL operators can be defined in terms of algebraic operations. An interpretation $\mathcal{I}$ is a pair $(U, .^{\mathcal{I}})$ where $U = \triangle^{\mathcal{I}}$ is the domain of interpretation and $.^{\mathcal{I}}$ the interpretation function. A concept C is interpreted as a set $C^{\mathcal{I}} \subseteq U$ and a role r as a binary relation $r^{\mathcal{I}}$ over U. In [4], a table representing the algebraic semantics of the expressive language $\mathcal{U}^-$ is presented.
The top and bottom concepts, the conjunction, disjunction and negation operators are defined as usually in DLs. The existential restriction is assigned to the *Peirce product*. Applied to a relation R and a set C, the Peirce product yields the set: $R : C = \{x \mid \exists y : (x, y) \in R \wedge y \in C\}$. The value restriction is assigned to a variant of Peirce product called *involution*. Namely: $(R : C')' = \{x \mid \forall y : (x, y) \in R \Rightarrow y \in C\}$.

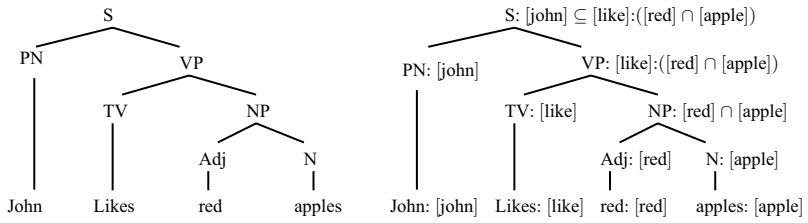

Figure 1: A sample derivation tree and its corresponding semantic tree

3.2 Natural language semantics and relational algebras

In [6] and later papers, Suppes uses relational algebras to achieve the semantic analysis of a fragment of the English language by annotating syntactic grammars with algebraic expressions.
The syntax of natural language is defined by a phrase structure grammar G, specified in terms of a set of production rules like those shown in the first column of Table 1.

	Lexical production rule	Semantic association
(i)	$S \rightarrow PN + VP$	$[PN] \subseteq [VP]$
(ii)	$NP \rightarrow N$	$[N]$
(iii)	$NP \rightarrow Adj + N$	$[Adj] \cap [N]$
(iv)	$VP \rightarrow TV + NP$	$[TV] : [NP]$

Table 1: semantic associations in relational grammars

The symbols S, NP, VP, PN, N, Adj and TV denote 'start symbol', 'noun phrase', 'verb phrase', 'proper noun', 'noun', 'adjective' and 'transitive verb' respectively.
Let U denote the domain of interpretation, U is a non-empty set. The denotation of phrases is given by a mapping [.] from syntactic types into an extended relation algebra $\mathcal{E}(U)$ over U. [.] is defined inductively by:

- A valuation on elementary types of the grammar G,
- Algebraic operations determining the denotation of non-elementary types.

The denotation of *elementary types* is defined by a partial function v, called a *valuation*. Elementary types are, for example, nouns and adjectives which v maps into sets in 2^U, transitive verbs that are mapped into binary relations in 2^{U^2}. Proper nouns are special elementary types, they are mapped to singleton sets.
The denotation of *non-elementary types* is defined by algebraic combinations from the denotations of elementary types. This is done by extending the phrase structure grammar with semantic functions associated to each production rule. Examples of semantic associations are shown in the second column of Table 1.
Let us now illustrate how meaning is assigned to a phrase by converting its grammatical definition to a semantical one. Consider for example the simple sentence:

John likes red apples (1)

We aim to find its adequate semantic representation. The syntactic structure of the sentence as defined by the phrase structure grammar of Table 1 is represented by the syntactic derivation tree of Figure 1.

The semantics is defined by a *semantic tree*. Figure 1 depicts the semantic tree for the sentence (1). It is constructed by assigning denotation to nodes starting from the leaves until the root is reached.

3.3 Translating the algebraic representation to a DL representation

To obtain the DL representation, words and phrases interpreted in the algebraic framework as sets are represented as concepts and those interpreted as binary relations are represented as roles. Given the exact correspondence between the algebraic operations and the DL constructs explained in Section 3.1, translating the algebraic representation to a DL representation is straightforward.
As subset relations correspond to subsumption relations and Pierce product to a some term, the terminological representation of the sentence (1) is then:

$$\text{John} \sqsubseteq \exists \text{like}.(\text{red} \sqcap \text{apple}) \tag{2}$$

Usually, systems used to describe natural language semantics are based on expressive terminological languages like $\mathcal{U}$ and $\mathcal{KL}$. Subsumption in such languages is undecidable. In our case, we are restricted to description logics with structural subsumption. Our natural language documents are represented by $\mathcal{L}_1$-terminologies. We assume that the terminologies obtained are acyclic.

4 Comparing terminologies

Let $\mathcal{L}$ be a description logic with a structural subsumption. Let $\mathcal{T}_1 = \{A_i \doteq C_i, i \in [1,n]\}$ and $\mathcal{T}_2 = \{A_j \doteq C_j, j \in [1,m]\}$ be two $\mathcal{L}$-terminologies. Extending the notion of best cover of a concept using a terminology to all the concepts occurring in $\mathcal{T}_1$, we define the difference between two terminologies as follows:

Definition 4 *(difference) Given a function ρ that maps every concept A_i occurring in $\mathcal{T}_1$ to its best cover using $\mathcal{T}_2$. The difference between the terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$ is the conjunction of the rests of all the concepts A_i occurring in $\mathcal{T}_1$ with respect to $\rho(A_i)$.*

$$diff_\rho(\mathcal{T}_1, \mathcal{T}_2) = \sqcap_{A_i \in \mathcal{T}_1} Rest_{\rho(A_i)}(A_i)$$

With the notion of size of a description, we define the dissimilarity coefficient between two terminologies.

Definition 5 *(dissimilarity coefficient) The dissimilarity coefficient between the terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$ is the size of their difference*

$$d(\mathcal{T}_1, \mathcal{T}_2) = |diff_\rho(\mathcal{T}_1, \mathcal{T}_2)|$$

Let us now illustrate our approach on an example. Consider the two following simple NL texts, describing the rooms of two motels:

Text 1	All the rooms are comfortable. Each room has a bathroom and contains a color TV.
Text 2	Each room has a bathroom and contains a phone. Only suites contain a color TV.

The corresponding terminologies must be normalized. First, we eliminate incomplete definitions. For each incomplete definition $A \sqsubseteq C$, a new atomic concept $\overline{A}$ is introduced, it stands for the absent part of the definition, we obtain: $A \doteq C \sqcap \overline{A}$. Second, the terminologies are unfolded, we substitute all defined concepts occurring in the right hand side of a definition by their definition.
For the terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$ we obtain:

$\mathcal{T}_1$	$\text{room} \doteq \text{comfortable} \sqcap \exists\text{have.bathroom} \sqcap \exists\text{contain.}(\text{color} \sqcap \text{TV}) \sqcap \overline{\text{room}}_1$
$\mathcal{T}_2$	$\text{room} \doteq \exists\text{have.bathroom} \sqcap \exists\text{contain.phone} \sqcap \overline{\text{room}}_2$ $\text{suite} \doteq \exists\text{contain.}(\text{color} \sqcap \text{TV}) \sqcap \overline{\text{suite}}$

Computing $diff_\rho(\mathcal{T}_1,\mathcal{T}_2)$ we obtain:
- $\rho(\text{room}) = \{\text{room} \sqcap \text{suite}\}$
- $diff_\rho(\mathcal{T}_1, \mathcal{T}_2) = \text{comfortable} \sqcap \overline{\text{room}}_1$
- $d(\mathcal{T}_1, \mathcal{T}_2) = 2$

Text 1 brings an additional information about the concept *room*, which is the fact that the rooms are comfortable.

5 Conclusion

We have considered the problem of comparing semantically two natural language texts. The first step of the work consists in translating natural language expressions into a formal representation. For that, we reuse the principles described in [5], that makes the connection between natural language semantics and description logics using relational algebras. We found that the notion of *best cover* can be used to compute the difference between two terminologies. The difference is computed by iterating the calculus of the best cover for all the defined concepts occurring in the first terminology.

The limit of our approach is the expressivity of the language since we are confined to description logics where the difference operation is semantically unique. Future work will be devoted to extend the method to more expressive DLs, overcoming this limit.

References

[1] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge: University Press, 2003.

[2] M. Boettner. Natural language. In C. Brink, W. Kahl, and G. Schmidt, editors, *Relationnal methods in computer science*, pages 226–246. Advances in Computing, Springer, Wien, 1997.

[3] M.S. Hacid, A. Leger, C. Rey, and F.Toumani. Dynamic discovery of e-services: a description logics based approach. In *18èmes Journées Bases de Données Avancées (BDA)*, pages 283–306, 21-25 Octobre, 2002.

[4] R. A. Schmidt. Algebraic terminological representation. Technical Report MPI-I-91-216, Max-Planck-Institut für Informatik, Saarbrücken, Germany, November 1991.

[5] R. A. Schmidt. Terminological representation, natural language & relation algebra. In H. J. Ohlbach, editor, *Proceedings of the sixteenth German AI Conference (GWAI-92)*, volume 671 of *Lecture Notes in Artificial Intelligence*, pages 357–371, Berlin, 1993. Springer.

[6] P. Suppes. Direct inference in english. In *Teaching Philosophy 4*, pages 405–418. 1981.