Identification of Textual Entailments in Business Rules

Erum Iftikhar, Anum Iftikhar
Dept.of Computer Science & IT
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
irum.iftikhar.mcs@gmail.com, anum.iftikhar01@gmail.com

Muhammad Khalid Mehmood
Directorate of IT
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
momikh@gmail.com

Abstract— **This paper presents an approach to automate the procedure of textual entailment recognition from business rules. Business rules are most important part of software requirements specifications, as little mistake in this phase results in absurd software design. Business rules are used in software industry. When we automatic translate these business rules we find entailment issue because business rule is not an independent sentence they have many module related to each other .so when we translate these business text we have found many issues such as discourse, semantic and negation problem. The evaluation method for business rules texts is to test against a list of sentences, each of which is paired with yes or no. For this case I have study business text and their problems in my MS thesis. What business rules are and what issues are found when we automatic translate these business rules. We used Stanford dependency parser for text translation.**

Index Terms— **Textual entailment, natural language processing, business rules**

I. Introduction

The aim of natural language processing (NLP) applications is to imitate human linguistic behaviour; however there is no agreement on how carefully it should be simulated. In such a complicated process, the device has to be conscious of the components used by the language as well as the global discourse and the perspective in which the dialogue is taking place. Therefore, apart from working with the information provided by the different linguistic analyses that an individual unknowingly creates, the program has to be able to merge these studies in the most appropriate way (Moreno et al., 1999).

A. Textual Entailment

Although in logics, entailment has official meaning, in pragmatics, this idea explains a merely a particular connection between two phrases or places of sentences: if the first is real, the other is also real. The regards between the linguistics field and application acting goes returning as far as the delayed 70's when Chen offered some heuristics to identify the elements of the Enterprise Relationship Design. He suggested that typical nouns generate entity kinds, transitive Spanish verbs connection kinds and adverbs features for relationships.

B. Business Process Text

Business process acting has become a vital device for handling business change and for catching specifications of application. As a result of competitors between organizations, improved customer objectives and unpredictable atmosphere, there has been a growing interest in new control paradigms that guarantee performance enhancement. In particular, paradigms concentrating on company procedure enhancement have drawn significant attention. This is because company procedures provide a useful structure to explain how work is achieved in a company. To evaluate and improve those procedures, several techniques have been recommended under the banner of Business Process Modelling (BPM). BPM contains many techniques, most designed to allow the discovery of the repercussions of implementing alternative procedure circumstances and styles. A comprehensive literary works has been created on this subject, most of it providing advice on how experts should go about it. Consultancies have been quick to create designed strategies, techniques and application programs to support BPM actions. At the same time, application organizations use BPM in the execution of many IT techniques, such as work-flow control, business source planning, and more lately, e-business techniques.

II. Background

In the previous section, it is outlined that how the use of natural language in business rules can affect the business process. Moreover, the software specifications specified in natural languages such as English are complicated to machine procedure. Indecisiveness and inconsistency in natural language specifications results in low precision in computerized application acting. The real inspiration behind this research study was to address this very important issue by reducing ambiguity in NL application specifications.

A possible remedy to get rid of ambiguity can be the use of a statistical (formal logic) reflection in place of natural languages to catch business process text. However, the use of logic reasoning for catching business process text is itself a complicated task and difficult to handle. Moreover, if somehow the specifications are taken in formal reasoning, it

978-1-5090-2000-3/16/$31.00 ©2016 IEEE

will be another serious issue to use statistical reasoning reflection at later stages of application growth such as application research, design, execution, examining, etc. Furthermore, share owners (such as clients) are typically not able to understand the mathematical reasoning. Hence, this remedy does not look possible.

A typical process in software development is software specifications. Business rules are generally taken by domain experts in natural languages. But a natural language such as English is syntactically uncertain and semantically unreliable. Hence software specifications written in English also became uncertain and unreliable. Indecisiveness and inconsistency in application specifications cause misunderstandings in later levels of application acting and growth. In natural languages such as English two types of complications may come across: syntactic ambiguity and semantic inconsistency. Indecisiveness in English language is a problem because the different visitors of the specifications requirements may understand different things.

III. Related Work

Information about the presence or absence of entailment has been useful for NLP tasks. This idea can also be effective for Machine Translation evaluation (MT) .Nevertheless; there are difference between TE and MT evaluation. First, TE assumes the text and hypothesis to be well defined sentences, which is not true in MT evaluation (Padó, S, .et.al. 2009).once we get a feature vector in machine learning approaches, classification procedure is easy. So, the primary concern is finding an appropriate feature space. The various possible feature spaces are: Entailment Triggers Features Distance Features and Pair Feature (Bhattacharya, A. 2012).However, it can be trained on more efficient pair wise preference judgments. Both designs can be used to estimate system level scores from phrase level scores. To evaluate and compare the performance of models, they use corpora that were designed by past instances of the WMT workshop. Results show that Comb work best and deliver a consistency of 0.53, competitive with the WMT 2008 results (Padó, S, .Et.al. 2009).

Machine translation engine is used to automatically generate English translation of the Japanese data. JAIST team takes part in three subtasks for Japanese: Binary class, RITA4QA and entrance exam. Cabocha tool were used for data pre-processing and for English translation they used Stanford core natural language processor. Cabocha is a SVM-based Japanese tool used for dependency parsing. Since June, 2001, it obtained the best performance (89.29%) among the mathematical parsing models for Japanese dependency parsing. The functions of Cabocha are: (1) Achieve high performance for dependency parsing based on SVMs (2) Use the PKE(ACL2003) to speed up the SVM classifier (3)Use deterministic cascaded chunking model and parse efficiently (4) C/C++/Perl/Ruby. However in this model, there were also restrictions such as: System is not accurate at discovering hard false entailment and System is not able to detect entailment relations Pham, Q. N. M., et.al. (2011).

The validation of the models has to depend on a discourse in NL (Leopold, H, .et.al.2012). In this paper, they address the issues of automatic verbalization for process Models. Their participation is a technique that is able to generate natural language Text from a process model, taking into account the problems of analysis labels. The sequencing of the process framework, as far as flexible phrases and texts planning. There has been increase in work during the last decade that is designed to offer efficient textual inference in unrelated websites about which the program has no skills. Such perform had happened best-known within the field of Question Answering (MacCartney, B, .et.al. 2006). The task of QA of finding solutions to natural language questions from huge text collections. The process of finding solutions needs processing texts in details that cannot be conducted at recovery time for very huge written text selections.

Recognizing Textual Entailment (RTE) is performed using Knowledge Management System (KMS). KMS is graphical interface that allows users to develop repositories of files and to perform a variety of tasks against the files. The tasks are performed using XML representation. KMS consist of three elements: 1) sentence splitter 2) sentence parser, 3) parse tree analyzer. the steps that are used to performed RTE tasks were: 1) parse the text and hypothesis & create XML representation (using KMS routines).repeat the steps for unparsed texts 2) evaluation of whether the texts entailed the hypothesis 3) compared the entities 4) show the results of the judgments and compute accuracy 5) extend the interface to more detailed analysis (examine results of subsets of full set)

The related research work discussed in this section shows that no prior work has done to detect entailments in SBVR business rules.

IV. Proposed Emotion Ontology

Entailment is widely used in many aspects of the human life. Assume that someone is seeking for something and he or she searches for the answer from books, friends, or the Web. In most cases, the information gathered or retrieved is not the exact answer, although the (information) seeker may have one in his or her mind. Instead, the consequences of the original goal may be detected, so the inference plays a role and confirms or denies the original information being sought. Entailment also occurs frequently in our daily communication, with respect to language understanding and generation. So it is important that we interpret the conversation according to situation. Although it is easier for the computers to interpret human dialogues.

First of all the question is why do we need meaning representation? The basic requirement of translating a sentence is to determine the relationship between the meaning of a sentence and the world as we know it. Second we implement a system that that compares the representation of the meaning of an input against the representation in its knowledge base. Here is important that system is able to compare the state of affairs described by a representation, to the state of affairs in some world as modelled by the knowledge base. Finally system transforms a natural language sentence into a logical formula.

The basic motivation behind the question: why it is important to extract entailment is that text application requires semantic inference and inference is done in an application dependent manner. So there is need a common framework for applied semantic. Textual entailment provides such framework (Jurafsky, D., et.al.2000).

A. *Steps of Textual Entailment Identification*

Following are main steps of identification of textual entailment in a NL statement:

1. Preprocessing
2. Representation
3. Knowledge sources
4. Control strategy
5. Justification

1) *Pre-Processing*

In this step we perform tokenization, stemming/ lemmatization and identify sop words.

Tokenization: Tokenization is the process of breaking a written text up into phrases, words, symbols, or other important elements known as tokens. The list of tokens becomes input for further steps such as parsing. Tokenization is useful both in IT & linguistics. Generally, tokenization happens at the phrase level. We use Stanford tokenizer for this step.

What counts as tokens in NLP? : First of all we defined what a token in NLP is. It depends on different language backgrounds. A token is

Methodologically useful

Linguistically significant

Steps of tokenization: Following are commonly used steps of tokenization.

Step 1: Segmenting Text into Words

Step 2: Handling Abbreviations

Step 3: Handling Hyphenated Words. Following are main types of Hyphens:

- End-of-Line Hyphen
- True Hyphen
- Lexical Hyphen
- Sententially Determined Hyphenation

Step 4: Numerical and special expressions.

Examples of numerical and special expressions are Email addresses, URLs, Complex enumeration of items, Telephone Numbers, Dates, Time, Measures, Vehicle Licence Numbers, Paper and book citations, etc.

English Enclitics: For our example a department of computer science can only be managed by an employee.

[A] [Department] [Of] [Computer] [Science] [Can] [Only] [Be] [Managed] [By] [An] [Employee] [.]

Here are 13 tokens for this text, and then we perform same on hypothesis.

Hypothesis who has $\geq$ 10yrs experience.

[Who] [Has] [$\geq$] [10] [Years] [Experience]

Here are 7 tokens for this text.

Stemming: In language and information retrieval, stemming is the procedure for decreasing terms to their root, base or stem form, usually a written text form. The stem need not be similar to the morphological base of the term; it is usually sufficient that related terms map to the same stem, even if this stem is not in itself a valid root. Stemming applications are known as stemming algorithms or stemmers. Algorithms used for this are Lookup algorithm, Suffix-stripping algorithm.

Successor variety stemmers: Successor variety stemmers are depending on linguistics which tried to figure out term and morpheme limitations in accordance with the submission of phonemes in a huge whole body of utterances. The stemming technique depending on this perform uses figures in position of phonemes, and a whole body of written text in position of phonemically transcribed utterances. Hafer and Weiss formally defined the technique as follows:

Let be a term of duration n; i, is a duration i prefix of. Let D be the corpus of conditions. Di is identified as the part of D containing those conditions whose first i figures coordinates i exactly. The successor wide range of i, denoted Si, is then described as the variety of unique figures that take up the i + 1st position of conditions in Di. A test term of duration n has n successor types S_1, S_2. S_n.

In less official conditions, the successor wide range of a series is the variety of different figures that follow it in conditions in some whole body of written text. Consider a whole body of written text made up of the following conditions, for example: able, axle, incident, ape, about.

To figure out the successor types for "apple," for example, the following procedure would be used. The first correspondence of the apple company is "a." "a" is followed in the writing whole body by four characters: "b," "x," "c," and "p." Thus, the successor wide range of "a" is four. The next successor wide range for the apple company would be one, since only "e" follows "ap" in the writing whole body, and so on.

When this procedure is performed using a huge whole body of written text (Hafer and Weiss review 2,000 conditions to be a constant number), the successor wide range of substrings of a term will reduce as more figures are added until a section border is achieved. At this point, the successor wide range will considerably increase. This detail is used to recognize arises.

Once the successor types for a given term have been produced, this detail must be used to section the phrase. Hafer and Weiss talk about four ways of doing this.

1. Using the cutoff technique, some cutoff value is chosen for successor types and a border is recognized whenever the cutoff value is achieved. The problem with this technique is how to select the cutoff value--if it is too small, wrong reduces will be made; if too huge, appropriate reduces will be skipped.
2. With the optimum and level technique, a section crack is created after a personality whose successor wide range surpasses that of the personality instantly previous it and the personality instantly following it. This technique eliminates the need for the cutoff value to be chosen.
3. In the finish term technique method, a crack is created after a section if the section is a finish term in the corpus.
4. The entropy technique uses the submission of successor wide range figures. The technique works as follows. Let |Di| be the variety of conditions in a written text whole

body beginning with the *i* duration series of figures. Let |Dij| be the variety of conditions in Di with the successor j. The possibility that a member of Di has the successor j is given by.

Using this formula, a set of entropy actions can be identified for a term. A set of entropy actions for forerunners can also be described in the same way. A cutoff value is chosen, and a border is recognized whenever the cutoff value is achieved.

Hafer and Weiss experimentally analyzed the various segmentation techniques using two criteria: (1) the variety of appropriate section reduces separated by the count of reduces, and (2) the variety of appropriate section reduces separated by the count of true limitations. They found that none of the techniques conducted completely, but that techniques that mixed certain of the techniques did best. The mostly used algorithm for stemming English, and one that has repeatedly been shown to be very effective, is porter algorithm Porter's algorithm includes 5 phases of terms reductions, used sequentially. Within each step there are various conventions to choose rules, such as choosing the rule from each rule group that applies to the longest suffix. In the first step, this convention is used with the following rule group

$\text{SSES} \longrightarrow \text{SS}$

$\text{IES} \longrightarrow \text{I}$

$\text{SS} \longrightarrow \text{SS}$

$\text{S} \longrightarrow \text{NULL}$

The Porter Stemmer is a conflation Stemmer designed by Martin Porter at the School of Arlington in 1980. The Stemmer is in accordance with the concept that the suffixes in the British terminology (approximately 1200) are mostly created up of a mixture of small and easier suffixes. This Stemmer is a straight line phase Stemmer. Particularly it has five actions implementing guidelines within each phase. Within each phase, if a suffix concept printed to a term, then the circumstances connected to that concept are examined on what would be the causing management, if that suffix was eliminated, in the way described by the concept. For example such a situation may be, the variety of vowel figures, which are followed be a consonant personality in the management (Measure), must be higher than one for the concept to be used.

Once a Rule goes its circumstances and is approved the concept shoots and the suffix is eliminated and management goes to the next phase. If the concept is not approved then the next concept in the phase is examined, until either a concept from that phase shoots and management goes to the next phase or there are no more guidelines in that phase whence management goes to the next phase. This procedure carries on for all five actions; the causing management being came back by the Stemmer after management has been approved from phase five.

The Porter Stemmer is a very commonly used and available Stemmer, and is used in many programs.

Stemming and Lemmatization with Python NLTK: Python NLTK can perform stemming as given in the following example.

$\text{T} \rightarrow$ A department of computer science can only be managed by an employee.

After stemming:

A depart of comput scienc can onli be manag by an employee.

$\text{H} \rightarrow$ who has ≥ 10yrs experience.

Who ha ≥ 10 yr experi.

Lemmatization: For lexical factors, texts are going to use different kinds of terms, such as arrange, arranges, and arranging. Furthermore, there are derivationally relevant terms with identical meanings, such as democracy, democratic, and democratization. In many conditions, it seems as if it would be useful for a look for one of these conditions to return information that contain another phrase in the set. The purpose of both lemmatization and stemming is to reduce related terms and texts to a root form.

However, the two terms vary in their meanings. Stemming usually represents a raw heuristic procedure. Lemmatization usually represents doing this with the use of morphological analysis of terms and vocabulary, normally trying to eliminate endings only and to return the base or vocabulary form of a term, which is known as the lemma. If experienced with the word ran stemming might come returning just r, whereas lemmatization would try to come returning either run or ran depending on whether the use of the symbol was as a action-word or a noun. The two may also differ in that stemming mostly breaks derivationally related terms, whereas lemmatization usually only breaks the different inflectional kinds of a lemma.

Identifying stop words: In computer science, stop words are terms which are filtered out prior to, or after processing of NLP text. A stop word may be recognized as a term that has the same like hood of occurring in those records not appropriate to a query.

2) Syntactic Representation

In this step texts are represent text through tree or graph and logical representation.

Syntactic parsing: In this step parse tree is generated and conveyed following information.

- POS tagging for each word
- Phrases
- Useful relation ships

Above mentioned steps are explained with the following example:

"A department of computer science can only be managed by an employee who has ≥ 10yrs experience"

Tagging: Here is an example of POS tagging performed using Stanford POS Tagger. `A/DT department/NN of/IN computer/NN science/NN can/MD only/RB be/VB managed/VBN by/IN an/DT employee/NN who/WP has/VBZ ≥/VBN 10yrs/JJ experience/NN`

Parse Tree Generation: Following is example of generation of a parse tree using the Stanford parser:

```
(ROOT
  (S
    (NP
      (NP (DT A) (NN department))
      (PP (IN of)
        (NP (NN computer) (NN science))))
    (VP (MD can)
      (ADVP (RB only))
      (VP (VB be)
        (VP (VBN managed)
          (PP (IN by)
            (NP
              (NP (DT an) (NN employee))
              (SBAR
                (WHNP (WP who))
                (S
                  (VP (VBZ has)
                    (VP (VBN ≥)
                      (NP  (JJ  10yrs)  (NN
experience))))))))))
    (. .)))
```

3) Knowledge Sources

The knowledge resources available to the system are the most important part of recognizing TE. In addition to syntactic parsing several techniques enhance the representation with various linguistics resources such as Stemming Predicate, POS tagging, argument representation (verb predicates and nominalization), Entity Annotation, resolution Dates, times and numeric values, identification and normalization, Event identification and Identification of semantic relations. Here semantic structural and pragmatic rules are defined. Usually WordNet is used for this purpose.

4) Control strategy

In this step we check that entailment occurs or not. We can do this with two methods. Either we compute similarity of text and hypothesis or use threshold value of text. The result occurs in binary form either entailment is or not.

5) Justification

In this step we proof our decision. The decision is not check by algorithm it can be evaluated using some proof e.g. using predicate calculus and lambda calculus reductions. First, the examples where entailments occur and it is assumed to be true,

T: The decision is made. _ H: The determination is made.

A determination & decision are judged with WordNet, so we have a rule all a (iq (a,'decison') <->iq (x,'determination')).

From above said scenario it is clear why the sentence were recognized as right entailments.

This chapter discusses in detail what textual entailment is and why we use it in natural language processing (NLP).textual entailment is a directional relation between two given text. This relation holds whenever the truth of one text fragments deduced from another text .different examples are given to explain the relation between entailment and presupposition. After that the necessary steps of textual entailment are discussed step by step with examples. All tools regarding to textual entailment also defined in this section.

V. Experiments and Results

In this step we identify textual entailment from business rules. This step is explained using 1st business rule and we consider this rule as text. In this step we calculate similarity using WordNet.

Example 5.1	If a rental request does not specify a particular car group or model, the default is group A
Text	If a rental request does not specify a particular car group or model, the default is group A
Hypothesis	If a lease demand does not specify a particular car group or design, the default is member A.

After filtering following terms are identified on which we recognized textual entailment.

Rental - lease
Request - demand
Model - design
Group - member

Rental – Lease =>∏ A (Iq(x,'Rental') -> iq(x,'Lease')).

Rental=>{02976952} <adj.pert>
Rental1#1 -- (available to rent or lease; "a rental car")
Pertains to noun {13077916} <noun.possession> rental#1 (Sense 1)
Rental =>{13077916} <noun.possession> **lease**#1, rental#1, letting#1 -- (property that is leased or rented out or let)
Entailment occur

Request – Demand =>∏ A (Iq(x,'Request') -> iq(x,'Demand')).

Request => {07086765} <noun.communication> request#2, asking#1 -- (the verbal act of requesting)..Hyporynym…**Ask** => 2. (433) ask -- (make a request or **demand** for something to somebody; "She asked him for a loan")
Entailment occur

Model – design =>∏ A (Iq(x,'Model') -> iq(x,'Design')).

Model=> {05766180} <noun.cognition> kind#1, sort#1, form1#2, variety1#5 -- (a category of things distinguished by some common characteristic or quality; "sculpture is a form of art"; "what kinds of desserts are there?") **Design**=>(8) design, plan -- (make a design of; plan out in systematic, often graphic form; "design a better mousetrap"; "plan the new wing of the museum")

Entailment occur

Group – Member =>∏ A (Iq(x,'Group') -> iq(x,'Member')).

Group=>{00029714} <noun.Tops> group#1, grouping#1 -- (any number of entities (members) considered as a unit)
Members=>=> {09672128} <noun.person> associate#1 -- (a person who joins with others in some activity; "he had to consult his associate before continuing")

Entailment occur

In this section results are calculated using above said case study and 3 more solved elsewhere, in which we calculated true entailment, false entailment and unknown entailment. Using these values we can calculate the value of recall, precision and F measure.

TABLE II: Results of the Experiments

Case studies	Total Entail-ments	True entail-ments	False entail-ments	Missed entail-ments	P	R	F-Measure
1	29	24	3	2	88.9	82.8	85.7
2	34	25	4	5	86.2	73.5	79.3
3	62	45	10	7	81.8	72.6	76.9
4	55	40	10	5	80.0	72.7	86.7
Avg.	45	33.5	6.27	4.76	84.2	75.4	82.1

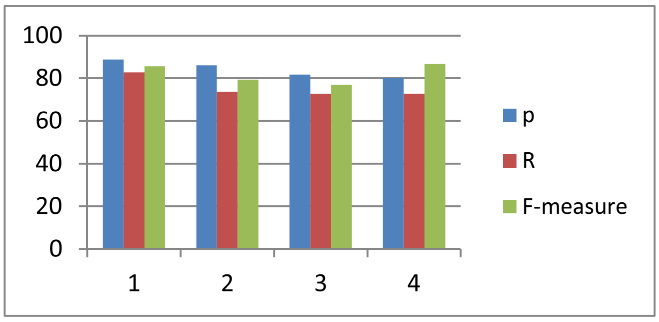

Fig. P, R and F-Measure values of all case studies

Case studies relating to business rules are solved in this chapter. All phases that are defined in figure 4.1 in chapter 4 are solved in these case studies one by one. Tables and graphs which are presented in this chapter show the effectiveness of our system. According to our results in table 5.1 show that Recall (84.2%) and Precision (75.4%) results applying on the used case study for business rule by using our system are very satisfactory. According to table 5.1 calculated F-Measure 82.1 is quite encouraging

VI. Conclusion

The main objective of this study was to improve the process of Textual entailment recognition from business rules by solving the ambiguity problem of natural Languages (NL).To deal with this task we have present a natural language based Stanford parser to parse English business rules and recognize textual entailment. Stanford parser is able to evaluate the piece of written text such as English which includes syntactic, lexical interpretation and semantic interpretation in its first phase and after getting propositions, our tool generates textual entailment from business rules in its last phase.

TReferences

[1] Adams, R. (2006). Textual entailment through extended lexical overlap. InProceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment (pp. 128-133).

[2] Akhmatova, E. (2005). Textual entailment resolution via atomic propositions. In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment (Vol. 150).

[3] Bosma, W., & Callison-Burch, C. (2007). Paraphrase substitution for recognizing textual entailment. In Evaluation of Multilingual and Multi-modal Information Retrieval (pp. 502-509). Springer Berlin Heidelberg.

[4] Dagan, I., Glickman, O., & Magnini, B. (2006). The pascal recognising textual entailment challenge. In Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment (pp. 177-190). Springer Berlin Heidelberg.

[5] Friedrich, F., Mendling, J., & Puhlmann, F. (2011, January). Process model generation from natural language text. In Advanced Information Systems Engineering (pp. 482-496). Springer Berlin Heidelberg.

[6] Glickman, O., Dagan, I., & Koppel, M. (2005). A probabilistic classification approach for lexical textual entailment. In PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (Vol. 20, No. 3, p. 1050). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[7] Jurafsky, D., Martin, J. H., Kehler, A., Vander Linden, K., & Ward, N. (2000).Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (Vol. 2). Upper Saddle River: Prentice Hall.

[8] Katrenko, S., & Adriaans, P. (2006). Using maximal embedded syntactic subtrees for textual entailment recognition. In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy (pp. 3-50).

[9] Kozareva, Z., & Montoyo, A. (2006). An approach for textual entailment recognition based on stacking and voting. In MICAI 2006: Advances in Artificial Intelligence (pp. 889-899). Springer Berlin Heidelberg.

[10] Leopold, H., Mendling, J., & Polyvyanyy, A. (2012). Generating natural language texts from business process models. In Advanced Information Systems Engineering (pp. 64-79). Springer Berlin Heidelberg.

[11] Malakasiotis, P., & Androutsopoulos, I. (2007). Learning textual entailment using SVMs and string similarity measures. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing (pp. 42-47). Association for Computational Linguistics.

[12] Marsi, E., Krahmer, E., & Bosma, W. (2007). Dependency-based paraphrasing for recognizing textual entailment. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing (pp. 83-88). Association for Computational Linguistics.