Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals of DAAAM International

TELLING COMPUTERS WHEN AND HOW TO ADAPT PROCESSES

TRANDABAT, D[iana] M[aria]

Abstract: *The next step in human-computer interaction is the use of human languages instead of some pre-defined commands. This paper intends to propose the use of natural language technologies in the management of industrial process flow. In order to make computers understand humans, language models need to be created from knowledge bases. We used such a knowledge base for the creation of a tool designed to analyse natural language semantics and determine* ***who*** *must perform* ***what*** *task* ***where when*** *etc. and we have the conviction that it can be used to make machines understand human when commanding the pipelining of processes.*
Key words: *Artificial Intelligence, Machine Learning, Tools*

1. INTRODUCTION

Every manufacturing process involves, at some step, the adaptation of parameters for a specific task. What if the rotation of a disk could be decreased simply by saying "*slow down one step*" or "*go slower*" instead of turning a button? Or if a pipeline of processes could be serialized by saying either "*Draw the whole before painting the product*" or "*Draw the whole after you're done painting the product*" instead of re-arranging the inputs and outputs of the two processes? Attracted by the potential applications, more and more researchers from the artificial intelligence field submerged into the natural language processing domain. Thus, one further step in the human-computer interaction is the use of human languages instead of some pre-defined expressions. In order to teach a computer to understand a human speech, language models need to be specified and created from human knowledge. While still far from decoding political speeches, computer scientists, electrical engineers and linguists have all joined efforts in making the language easier to be learned by machines, and with the results obtained so far in the field, applications as the upper ones are very realistic.

The paper is structured into 7 sections. Section 2 presents the trends in the natural language processing field, especially in the semantic analysis. Section 3 describes a natural language analysis system for several languages, while the results of the systems are presented in Section 4. The presentation concludes with a summarizing discussion in Section 5, including some further considered developments. The financial support is acknowledged before the final Reference section.

2. ANALYSING NATURAL LANGUAGES

Once the operator has spoken (or written) its commands, the machine starts to process them: identify all the words, ignore any non-essential communication-driven words (such as "*and now, we're going to*" or "*euh... what next?*"), identify the role players ("*we*", "*you*", "*button*", "*product*", etc.) and attach syntactic information, such as negation ("*decrease*" vs. "*don't decrease*") or tense information ("*latter, we will close the valve*" vs. "*close the valve*"). While this seems (and is) a very complex process, depending on the performances of the natural language processing system, it can take up to a few seconds.

All the tasks enumerated above are examples of natural language processing systems that need to be adjusted before the machine will understand the command. Thus, a key concern in the natural language processing domain is the identification of the mechanism that allows the attachment of meaning to texts, in order to re-create it in the machine's brain. These are the relations between ***semantic roles*** and an ***action verb***.

The natural language processing community has recently experienced a growth of interest in semantic roles, since they describe the role players in texts: WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW etc. for a given situation. For example, if the operator's command would be "*close the valve in the left corner*", the semantic roles that occur in this command, required by the command verb *close*, are: the object *the valve* and the place/position *in the left corner*.

A fixed set of semantic relations is not yet defined, but starting with (Fillmore, 1968), followed by (Frawley, 1992; Jackendoff, 1990), to mention just a few, different number of roles have been used. In recent years, a number of studies, such as (Chen and Rambow, 2003; Gildea and Jurafsky, 2002), have investigated the task of automatically identifying semantic roles in texts. Role assignment has generally been modelled as a classification task: a statistical model is created using manually annotated data and later used to assign a role label out of a fixed set to every semantic role of the predicates in a new, unlabeled sentence. The best results were reported with SVMs (Pradhan et al. 2005), with a highly optimized feature set. This paper presents a similar system, PASRL, build for identifying semantic roles in texts, intended to be used in analyzing the commands in manufacturing processes.

3. PASRL

In order to identify which of the machine learning techniques is best suited for the semantic role identification task (SRL), we have tested several techniques for different languages, using the algorithms implemented in the Weka toolkit (Witten & Frank, 2005). The final system developed into a platform for creating supervised Semantic Role Labelling systems. The platform trains several classifiers, chooses the ones with the greatest performance and returns a Semantic Role Labelling System which will be used to annotate new data. This returned system can be used to assign semantic roles to the commands in the industrial customization processes.

The training data used for the development of PASRL was the training and development resource from the ConLL 2009 Shared Task (Surdeanu et al., 2008), consisting of manually annotated treebanks such as the Penn Treebank for English, the Prague Dependency Treebank for Czech and similar treebanks for Catalan, Chinese, German, Japanese and Spanish languages, enriched with semantic relations (such as those captured in the Prop/Nombank and similar resources). For Romanian, a semantic role resource was automatically created, starting from the English resource (Trandabat, 2007), and used as training corpus.

Our system is composed of two main sub-systems: A *Predicate Prediction* module and an *Argument Prediction* module. The Predicate Prediction module has two possible configurations, corresponding to the *Predicate Identification* and *Predicate Sense Identification* layers. The first configuration involves a sequential identification of the predicates in a sentence, followed by the assignment of the predicate sense. The second configuration allows for a joint learning of the predicates in sentence, together with their senses.

The second sub-system, the *Argument Prediction* system, performs argument prediction based on the dependency relations previously annotated with the MaltParser (Nivre, 2003) and the predicate senses (the output of the *Predicate Sense Identification* module).

For each problem, the modules have three variants, related to the training set size. Training the whole system for a particular language requires running tens of classifiers, therefore running the classifiers on the whole training size is a very time expensive task. Therefore, the training data has been filtered and, besides running the classifiers on the whole data size, we offer the possibility to train, for each problem, different classifiers for the noun phrase or verb phrase or even more refined, for each noun, verb respectively in the training set.

For each module, a set of classifiers from Weka framework are trained. After running all the classifiers for all the modules, their performance is compared, and the path that obtains the highest performance is considered the best configuration. The models for this best configuration are saved, and the best path is written to a configuration file. This configuration can then be used at a later time to annotate new texts with PASRL. If all the created models are saved, and not just the best performing ones, the user can define, using the configuration file, the sequence of classifiers it wishes to run for each subtask to annotate new texts using the pre-trained models

4. RESULTS

The evaluation of the PASRL performance was computed using 10-fold cross-validation on the training set. For each task, PASRL evaluates all the machine learning algorithms used against the gold-annotated corpus, and the best performing algorithm is saved for the configuration file (more details in Trandabat, 2010). The evaluation was performed considering the number of correctly classified labels. PASRL was tested using the training data for different languages, available through the ConLL 2009 Shared Task. When considering the 5 best classifiers for the Predicate Identification task evaluated on the whole training data, using 10-fold cross-validation, we can observe that SimpleCart, J48 and ClassificationViaRegression are among the 5 best algorithms for different languages, which suggests that semantic role labelling may not be very language dependent, once the input text is syntactically annotated, which seems natural since semantic roles are representations at the deep, conceptual structure of the language.

Another important observation is that the best performing algorithms score differently for different languages. For instance, the best performances for Czech, German and Chinese range around 97-98%, while for English the best performance is 86% and for Japanese only almost 80%. Since the size of the training corpus is similar, one possible explanation could be the different level of inflection, the free vs. fixed word order, or the position of the verb with respect to the other elements of the sentence, or a combination thereof.

When evaluating the pre-trained models for English on new data, using the whole processing chain (including part of speech and dependency annotation), the results are promising, with 68% for noun predicate and 81% for verb predicate F1.

5. CONCLUSION

A large part of the work done in NLP deals with exploring how different tools and resources can be used to improve performance on a task. The quality and usefulness of the resource certainly is a major factor for the success of the research, but equally so is the creativity with which these tools or resources are used.

Recognizing and labelling semantic arguments is a key task for answering *Who* should do *what, when, where, why,* etc. Having built the semantic role system, we can now use it in various applications. The most immediate application will be in interpreting short messages, commands or comments from operators, in order to assist the human experts in performing parameters adjustments in manufacturing processes. With this paper, we intend to attract interested possible users in order to analyse some user scenarios.

A second direction will be to use this system in message generation applications, by allowing the machine to respond to the comments of operators, by adding some frequent commands to the default list, for instance.

We believe that, by joining natural language processing and process control flow, a new computer assisted process management direction can be instituted.

6. ACKNOWLEDGEMENTS

The research presented in this paper was funded by the Sectoral Operational Programme for Human Resources Development through the project "Development of the innovation capacity and increasing of the research impact through post-doctoral programs" POSDRU/89/1.5/S/49944.

7. REFERENCES

Chen J. and O. Rambow. (2003) Use of deep linguistic features for the recognition and labeling of semantic arguments. *Proc. of the 2003 Conf. on EMNLP* 2003

Fillmore Charles J. (1968) The case for case. In Bach and Harms, editors, Universals in Linguistic Theory, pages 1-88. Holt, Rinehart, and Winston, New York, 1968

Frawley W. Linguistic Semantics. (1992) Hillsdale, NJ: Lawrence Erlbaum Associates., 1992

Gildea Daniel and Daniel Jurafsky. (2002) Automatic labeling of semantic roles. Computational Linguistics, 28(3):245-288, 2002

Jackendoff Ray. (1990) Semantic structures. MIT Press, Cambridge, Massachusetts, 1990.

Nivre J. (2003) An efficient algorithm for projective dependency parsing. *Proc. of the 8th Intl Workshop on Parsing Technologies (IWPT 03)*, pages 149-160, 2003

Sameer Pradhan, Kadri Hacioglu, Valeri Krugler, Wayne Ward, James H. Martin, & Daniel Jurafsky. (2005). Support vector learning for semantic argument classification. *Machine Learning Journal*, 60(13):11-39, 2005

Surdeanu Mihai, Richard Johansson, Adam Meyers, Lluis Marquez, & Joakim Nivre. (2008) The CoNLL-2008 Shared Task on joint parsing of syntactic and semantic dependencies. *Proc. of the 12th Conf. on Computational Natural Language Learning (CoNLL-2008)*, 2008

Trandabat Diana (2010) Natural Language Processing Using Semantic Frames, PhD Thesis, University Al. I. Cuza Iasi, Romania, 2010

Trandabat Diana. (2007) Semantic frames in Romanian natural language processing systems. *Proc. of the NAACL-HLT 2007 Doctoral Consortium*, pages 29-32, Rochester, New York, 2007. Association for Computational Linguistics

Witten Ian H. & Eibe Frank. (2005) *Data Mining: Practical Machine Learning Tools and Techniques* (Second Edition). Morgan Kaufmann, 2005